Journal of AI Application

Volume 1, Issue 1
October 2024

Journal of AI Applications

Volume 1, Issue 1 October 2024

Contents

Quantitative Representation Learning of Community Life Circle Structure

Zheng Shiyu Liu Junling

School of Computer Science and Engineering, Shenyang Jianzhu University, Shenyang 110168, Liaoning, China

***Abstract*—**The construction of community living circles is an important component of urban community planning. Reasonably determining the necessary POIs within the living circle, discovering the connection between community residents and POIs, and quantifying the spatial structure of the community living circle can provide effective basis for creating a livable living environment. We propose a quantitative representation method for the structure of community life circles. Based on representation learning technology combined with urban residents' activity patterns, the accurate relationship between life circles and POI is discovered, solving problems such as determining the scope of existing community life circles and blurring dependency relationships. Firstly, unlike existing methods for defining fixed range living circles, we use k-nearest neighbor (kNN) to determine candidate POIs in community living circles; Then, a method for calculating the quality index of POI was designed, and the dependency index of the community on each POI was computed using the POI quality index and the category weight of POI. The time period was divided using the KL divergence measure to construct activity maps that reflect the travel patterns of residents at different time periods; Finally, an automatic encoder is used to embed the activity graph for representation learning, and a feature vector representation of the community life circle structure is obtained. For the convenience prediction application, experimental evaluations were conducted using real datasets to verify the effectiveness of the proposed quantitative representation method for community life circle structure.

***Keywords*—**kNN; POI quality; community life circle; community life circle structure; representation learning

I. INTRODUCTION

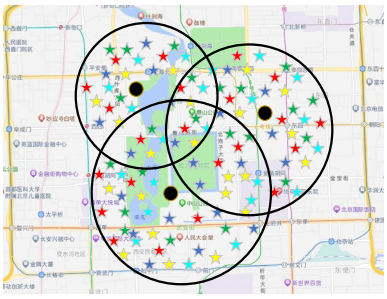
China's urbanization has seen significant improvement, and the construction of urban living circle service facilities plays a crucial role in enhancing residents' quality of life. In July 2021, the Ministry of Commerce and ten other departments issued the "Guidelines for the Construction of 15-Minute Urban Convenience Living Circles," outlining a plan to establish "hundreds of cities with thousands of living circles" by 2025, aiming to develop a series of reasonably laid out and fully functional urban convenience living circles [1]. Points of Interest (POIs), as important forms of urban spatial facilities, possess good representation capabilities of urban morphology and serve as a vital basis for exploring urban spatial structures. POI-based research on urban structures has yielded a series of achievements in the fields of urban infrastructure planning, commercial space layout, and residential distribution patterns. Meanwhile, with the rapid development of information technology, residents' spatio-temporal behaviors increasingly influence the organization and structure of urban space, making in-depth analysis of urban space and residents' spatio-temporal behaviors highly valuable for research [2]. By combining the patterns of human activities in space, discovering the hidden patterns in these data can help better understand urban spatial structures.

The spatial coverage of community living circles and the allocation of public service resources significantly impact the spatial behaviors of community residents [3]. Existing research on community living circles includes the delineation of community living circle boundaries and the discovery of

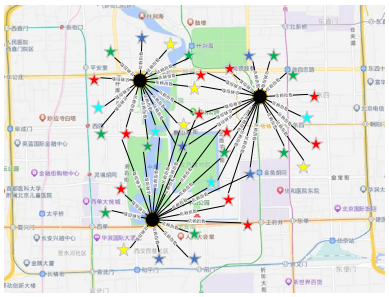
community living circle structures [4]. Methods for delineating community living circles focus on the role of temporal and spatial scales, measuring the boundaries of community living circles based on facility accessibility either in fixed time or length, without considering the quality and scale factors of POIs themselves [5-6]. However, POI quality and scale have a major impact on community contributions. For example, some large-scale and well-serviced POIs can influence distant communities and be shared by multiple communities. Conversely, certain nearby POIs that are small-scale and provide average services have limited influence on community residents.

Research on community living circle structures is mainly divided into two categories: one explores the spatial allocation of urban service facilities by analyzing the distribution of static POIs [7], and the other identifies urban regional functions and explores inter-regional differences by analyzing the activity patterns of community residents from a macro perspective [8-10]. These methods do not consider the impact of POI quality and do not focus on fine-grained structural analysis of communities. For instance, within a community living circle, some large-scale and well-serviced POIs, even if they are far from the community, may be more reliant on by community residents than nearby POIs that are small-scale and provide average services. Therefore, delineating community living circles more in line with residents' actual living patterns and quantitatively representing the structure of community living circles in fine granularity based on the distribution of surrounding facilities can help better understand the current status of community living circles and improve the layout of facilities around communities.

This paper proposes a quantitative representation method for community living circle structures, utilizing representation learning techniques combined with urban residents' activity patterns to discover accurate relationships between living circles and POIs. Figure 1 provides an example of the structure of community living circles. Figure 1(a) depicts the spatial distribution of community living circles, with three communities surrounded by various types of POIs such as restaurants, hospitals, schools, and entertainment venues. The scope of each community living circle varies in size, and there are overlaps among them, with shared POIs existing in the overlapping areas. Figure 1(b) illustrates an example of the obtained activity graph of community living circles, represented as a bipartite graph structure. There are two types of nodes in the graph: community nodes and POI nodes. Each community is connected to its essential POIs through dependency relationships, and the activity graph of community living circles showcases the essential POIs surrounding each community and the degree of dependency of the community residents on these POIs.



(a) The spatial distribution of community living circles



(b) Activity graph of community living circles

Fig. 1 Example of community life circle structure

Constructing an effective structural diagram for urban community living circles is challenging, mainly reflected in the following aspects: (1) A large number of communities and POIs are distributed across urban spaces, and there are shared relationships among different communities regarding POIs. Directly delineating the scope of living circles based on straight-line distance or walkable distance within a certain time frame does not align with residents' actual living situations. Determining the essential POIs within community living circles accurately and reasonably poses a challenge. (2) Each community living circle is populated with numerous POIs, and merely counting the number of POIs or the frequency of residents' visits to them cannot accurately describe the structure of the community living circle. Quantifying the importance of POIs to residents' daily lives is a challenge. (3) Residents' activities within living circles exhibit periodic dynamic changes, and effectively capturing these dynamic structural features is a challenge.

To address these issues, we first use the k-nearest neighbors (kNN) method to identify essential POIs within living circles and, after establishing connections between communities and POIs, select some shared POIs among communities based on distance factors. Then, we quantify the quality index of POIs by combining POI scale data and review data, and on this basis, calculate residents' dependency index

on various types of POIs by incorporating POI category weights. Finally, we use a method based on topic modeling and sliding windows with KL-divergence as a metric to divide each day into time periods and perform collective embedding representation learning on the community living circle activity graphs for each time period, obtaining a vector matrix that can effectively represent the dynamic structural features of community living circles.

The main contributions of this paper are as follows:

(1) We propose a method to determine essential POIs in community living circles, which is used to define the scope of community living circles, and define a quantitative method for measuring the influence of POIs on communities to describe residents' dependency on POIs within living circles.

(2) Using representation learning methods, we obtain fine-grained dynamic structural features of community living circles and obtain a vector matrix that can effectively represent the structural features of community living circles.

(3) For applications such as predicting the convenience of community living circles, we conduct experimental evaluations using real datasets and verify the effectiveness of the proposed quantitative representation method for community living circle structures.

II. INTRODUCTION

This section introduces related work on urban functional area division, community living circle structure, and representation learning.

A. Urban Functional Area Division

With the rapid development of urban construction and the continuous maturation of sensing technology and computational environments, urban computing based on urban big data has received widespread attention. In the field of urban spatial data computing, the increasing popularity of internet electronic map services and LBS applications has made the computation of community structure and function based on urban public service facilities an important research direction in urban computing [11].

Research on functional area identification mainly relies on POI data to determine the main functions of areas based on user behavior and subsequently divide static urban functional areas. Reference [12] uses the skip-gram model and t-SNE technology to explore and visualize urban area functions using urban POI data. Reference [9] utilizes taxi trajectory data to obtain residents' visits to various POIs, thereby discovering urban area functions. Reference [10] proposes a probabilistic latent factor model and uses it to learn the mixed functions of urban areas.

B. Community Living Circle Structure

The division of the spatial scope of community living circles has a significant impact on the structural analysis of living circles. Existing methods for dividing the spatial scope of living circles include: using Euclidean straight-line distance [7], walkable range within a certain time [13-14], delineation considering population size [15], area division [16], and methods considering factors such as service facility convenience [17]. With the increasing popularity of big data technology, Reference [18] proposes the use of GPS and mobile phone signaling data to collect information on the actual activities of community residents to achieve a more refined division of living circles. However, due to the diversity and complexity of GPS behavior subjects, a large sample size survey is required.

This paper does not fix the travel time or distance for delineating the scope of community living circles but instead uses the kNN method to determine the essential POIs for community living circles.

In terms of structural representation, Reference [19] uses POIs and their street view images to describe community structure; Reference [8] uses transportation data to obtain residents' visit frequencies to various POIs within living circles and, on this basis, quantitatively represents the mobility connectivity between communities and POIs, thereby establishing a community structure graph.

This paper focuses on the impact of POI quality on community residents, establishes activity graphs reflecting the structure of each community living circle by quantitatively representing the dependency relationship between community residents and POIs. Based on this, the activity times of community residents are periodically divided, and representation learning methods are used to obtain the dynamic structural characteristics of community living circles over time.

C. Representation Learning

Representation learning is a learning method that automatically discovers implicit features in data. Compared to feature engineering methods in machine learning, representation learning avoids the process of manual feature extraction.

Representation learning is divided into text representation learning and graph representation learning [20]. Graph representation learning aims to obtain a low-dimensional

feature vector through learning to represent the graph structure with high-dimensional data [20]. Graph representation learning can be divided into probabilistic graph model algorithms, manifold learning algorithms, and reconstruction algorithms based on autoencoders. This paper selects the reconstruction algorithm based on autoencoders, which projects the implicit features in the original data into a low-dimensional feature space through a nonlinear mapping function, minimizing the loss between the original and reconstructed features [21]. Compared to the other two algorithms, the reconstruction algorithm based on autoencoders uses latent variables and does not require complex posterior distributions. The autoencoder can directly parameterize features or representation functions and learn a feature vector that directly encodes them [22-23].

By conducting collective embedding representation learning on periodically changing community living circle activity graphs, this paper obtains a vector matrix that can effectively represent the structural characteristics of community living circles.

III. Problem Definitions

Definition 1. Candidate POIs for Community Life Circle. Given a community c_k and a POI set P_0, the set of candidate POIs within the life circle of c_k is denoted as P, which comprises all POIs in P_0 that are within a distance not exceeding the k-th nearest neighbor of c_k in P_0. Formally, P = {p ∈ P_0 | dist(c_k, p) ≤ dist(c_k, p')}, where p' is the k-th nearest neighbor of c_k in P_0.

Utilizing the kNN method to identify candidate POIs within a community life circle allows for the screening of POIs that are closer in distance and have practical relevance to residents among the numerous POIs surrounding the life circle.

Definition 2. Dependency Relationship Between Community and POI. Given a community c_k and a set of candidate POIs P for its life circle, E represents the set of dependency relationships between c_k and P. Each relationship in E is represented as a triplet e(c_k, p_j, w), where w is the weight between c_k and p_j, calculated as $score_j * D_s$. Here, $score_j$ is the quality index of POI p_j, and D_s is the weight of the POI category in which p_j belongs.

In daily life, the dependency of community residents on POIs within their life circle is influenced not only by the distance between them but also by the quality of the POIs themselves. This paper emphasizes the consideration of POI quality when quantifying the relationship between communities and POIs.

Definition 3. Activity Graph of Community Life Circle. Given a community c_k, the activity graph for a single time segment of its life circle is denoted as $G_k(P, E)$, where P is the set of candidate POIs within the life circle of c_k, and E is the set of dependency relationships between c_k and P. Any relationship e(c_k, p, w) ∈ E indicates that p is a necessary POI within the life circle of c_k that has a dependency relationship with c_k. The activity graph of a community life circle is a collection of single-time-segment activity graphs, formally represented as $G^k = \{G_1^k, G_2^k, G_3^k, \ldots, G_t^k, \ldots, G_n^k\}$, G_t^k represents the activity graph of community c_k 's life circle during the t-th time segment.

The interaction between community residents and POIs always exhibits dynamic changes over time. For example, people tend to visit shopping malls less frequently on weekdays than on rest days; in the mornings of weekdays, they may be more active at bus stops or subway stations, while in the evenings, they may be more likely to appear at restaurants or squares. Therefore, to better describe the time-varying mobility of community residents, this paper constructs periodic multi-time-segment activity graphs for community life circles.

Definition 4. Embedding Representation Learning of Community Life Circle Activity Graph. Given a community life circle activity graph , $G^k = \{G_1^k, G_2^k, G_3^k, \ldots, G_t^k, \ldots, G_n^k\}$ by learning a spatial mapping function $f(c_k): G^k \rightarrow R_k^d$, the structural features in each activity graph of the community life circle are mapped into a vector representation R_k^d.

IV. Methods

This section introduces the method for quantitative representation learning of community life circle structures. Firstly, the overall framework structure is presented; subsequently, three key steps are elaborated in detail: the construction of a single-time-period community life circle activity graph, the construction of a multi-time-period community life circle activity graph, and the representation learning of quantitative structures within community life circles.

A. Construction of Community Life Circle Activity Graphs

In constructing the community life circle activity graph for a single time period, this paper adopts the following strategy. Firstly, the quality index of Points of Interest (POIs) is calculated using their review data and size data. Then, a community life circle activity graph construction algorithm is proposed, which utilizes the k-Nearest Neighbors (kNN) algorithm to determine candidate POIs within the community life circle. By combining the quality index with the category weight of POIs, a dependency index for each POI within the community is synthesized, thereby establishing a connection between them.

(1) Generation of POI Quality Index

This paper quantifies the influence of POIs by calculating their quality index. The quality index is measured based on the number of reviews reflecting POI popularity, the rating score, and the occupied area reflecting facility size. The review data is sourced from a popular review website, while the area data is obtained from the Area of Interest (AOI) dataset. Since there is not a one-to-one correspondence between AOI data and POI data, such as multiple shops within a mall, each with POI point information on the map but not AOI polygon information, the quality index calculation for such POIs only considers the impact of the number of reviews and rating score.

When synthesizing the POI quality index, due to the different dimensions of the three influencing factors (number of reviews, rating score, and occupied area), normalization of the indicators is first required. Since the distribution of each indicator is not a normally distributed real number and is non-negative, the Min-Max normalization method is chosen for normalization, scaling the three influencing factors uniformly between 0 and 1. The specific normalization formula is shown in Equation (1).

$$X_{j,i}^{*} = \frac{X_{j,i} - \min(X_{*,i})}{\max(X_{*,i}) - \min(X_{*,i})} \quad (1)$$

Where $X_{j,i}^{*}$ is the normalized value of the i-th influencing factor for the j-th POI's quality index, $\min(X_{*,i})$ is the minimum value of the i-th influencing factor for all POI quality indices, and $\max(X_{*,i})$ is the maximum value. After normalization, the three indicators are weighted and summed to obtain the quality index $score_j$ for each POI. The specific calculation formula is shown in Equation (2), where $weight_i$ is the weight of the different influencing factors for the POI quality index.

$$score_j = \sum_{i=1}^{3} weight_i * X_{j,i}^{*} \quad (2)$$

The determination method for the weights (\(weight_i\)) of the influencing factors of the POI quality index adopts the Analytic Hierarchy Process (AHP). The AHP adopts a consistent matrix method, comparing all influencing factors pairwise to minimize the comparison difficulty due to the different natures of the factors, thereby improving accuracy. This paper conducts pairwise comparisons among the three influencing factors of the quality index and assigns qualitative ratings based on their importance. The matrix formed by the pairwise comparison results is called the judgment matrix A, with elements a_{ij} representing the comparison results of the importance of factor *i* and factor *j*. When factors *i* and *j* are equally important, the comparison result is 1. With increasing importance, four levels are assigned with values of 3, 5, 7, and 9, with intermediate values of 2, 4, 6, and 8 for each level. The judgment matrix has the following property: $a_{ij} = \frac{1}{a_{ij}}$. After passing the consistency test, the obtained judgment matrix A for influencing factors can be used to determine the corresponding weights of the three influencing factors of the quality index using the eigenvalue method. The consistency test formula is shown in Equation (3).

$$CI = \frac{\lambda_{max} - n}{n-1} \quad (3)$$

Where CI is the consistency test result, λ_{max} is the maximum eigenvalue of the judgment matrix A, and *n* is the order of the judgment matrix A. If CI = 0, it proves that the judgment matrix A has complete consistency; if CI is close to 0, it proves that there is satisfactory consistency, and the consistency of the judgment matrix A can be accepted. The larger the CI value, the more severe the inconsistency of the judgment matrix A, which then requires revision.

Considering that different age groups have different judgments on the importance of the same POI influencing factor, this paper introduces an age group judgment matrix to eliminate the impact of different ages on the weight generation results, thereby improving accuracy. This paper divides age groups into four stages: juvenile, youth, middle-aged, and elderly. The calculation formula for the weight $weight_i$ of POI quality influencing factors, incorporating the results of the age group judgment matrix, is shown in Equation (4), where h_j is the weight of each age group, and v_{ij} is the judgment result of the importance of the i-th influencing factor by the j-th age group.

$$weight_i = \sum_{j=1}^{4} h_j v_{ij} \quad (4)$$

(2) Construction Algorithm for Community Life Circle Activity Graph

In this paper, the kNN method is employed to identify candidate POIs within the community life circle. By calculating the dependency index of the community on different categories of POIs, the connection between the community and the POIs within its life circle is established. Based on this, a community life circle activity graph for a single time period is constructed. The activity graph construction algorithm is shown in Algorithm 1.

Algorithm 1 Construction Algorithm for Community Life Circle Activity Graph

Input: Community c_k ,urban POI set P;
Output: Community life circle activity graph $\boldsymbol{G_t^k}$ for a time period.

(1) **for** each POI point p_j in the urban POI set;
(2) $score_j = \sum_{i=1}^{3} weight_i * X_{j,i}^{*}$;
(3) Use kNN to determine the number of candidate POIs of each category in c_k;
(4) **for** each category s of candidate POIs in the community c_k life circle

(5) **for** each candidate POI point p_j in each category
(6) **if** (p_j has not share community)
(7) $w_{kj} = score_j D_s$;
(8) **else**
(9) $w_{kj} = \frac{1}{H_{kj}} score_j D_s$;
(10) **if**(w_{kj}<average) delete w_{kj}
(11) **return** G_t^k;

Given community c_k and the urban POI set P, the algorithm mainly consists of three steps. In Step 1, the quality index of all urban POIs is calculated (lines 1-2). In Step 2, kNN is used to determine the candidate POIs within the community life circle (line 3). In Step 3, the connection between community c_k and the candidate POIs within its life circle is established (lines 4-10). The calculation of the POI dependency index is divided into two cases: shared by multiple communities and not shared (lines 6-9). Here, w_{kj} represents the dependency index of community c_k on candidate POI p_j within its life circle, $score_j$ is the quality index of p_j, D_s is the weight of the POI category where p_j belongs, and H_{kj} represents the distance between community c_k and POI p_j. When the w_{kj} assigned to a POI shared by multiple communities is less than the overall average, that edge will be pruned (line 10). Finally, the community life circle activity graph G_t^k for a time period is output (line 11).

Table 1 Values for the number of POIs in 8 categories

	Shop	school	Medical	Catering	Leisure	Transportation	Daily	Culture
num	36	5	20	70	20	5	40	10
weight	0.3	0.1	0.06	0.2	0.04	0.2	0.06	0.04
num	36	5	20	70	20	5	40	10
weight	0.3	0.1	0.06	0.2	0.04	0.2	0.06	0.04

Medical Services, Catering, Leisure, Transportation, Daily Life, Culture and Sports

The kNN (k-Nearest Neighbors) query for spatial point data involves finding, within a given two-dimensional spatial dataset, the set of all data points whose distances to the query point do not exceed the distance to the k-th nearest neighbor. In real life, people tend to visit the nearest POIs (Points of Interest), therefore, this paper selects the kNN method to determine the number of necessary POIs, and the experimental results validate this approach. Regarding the selection of the k value, this paper refers to the conclusions on k-nearest neighbor values for various facilities within the community life circle from literature [24], as shown in Table 1.

The calculation formula for the POI dependency index adopts the dimension-free multiplication method proposed in literature [25]. For the POI category weights, this paper refers to the POI weight values related to resident convenience summarized in literature [26] to determine the values of D_s, as shown in Table 1.

Figure 3 presents an example of a community life circle activity graph. As shown in the figure, except for p_1, all other POIs are contained within only one of the communities c_1, c_2 and c_3. Therefore, the dependency index w_{kj} between them can be directly calculated by multiplying $score_j$ with D_s. However, p_1 is shared by all three communities c_1, c_2 and c_3, and the dependency index of p_1 is allocated to each community in inverse proportion to the distance between the community and p_1. Meanwhile, when the distance between a shared community and a POI is too far, the allocated dependency index will be very low, indicating a weak connection between the community and the POI. Therefore, when the dependency index of a shared community to a POI is lower than the urban average for that facility category, the edge between them is removed.

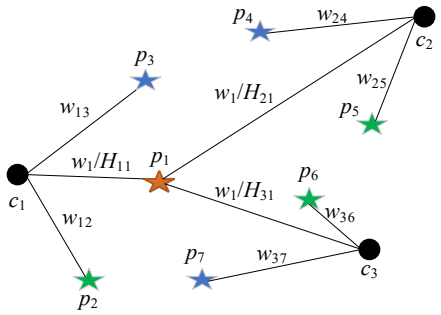


Fig. 3 Example of community life circle activity graph

B. Construction of Multi-time-period Community Life Circle Activity Graphs

To better capture the dynamic structural characteristics of community residents' activities over time, this paper periodically divides the time periods for constructing community life circle activity graphs. In terms of time period division methods, literature [8] generated seven dynamic graphs based on data from Monday to Sunday. To obtain finer-grained time segments, this paper adopts the method proposed in literature [27], which utilizes a sliding window approach and measures based on KL-divergence for segmenting different time periods of each day, as shown in Equation 5. Compared to Wesserstein distance and JS divergence metrics, KL-divergence is more suitable for evaluating fine-grained and overlapping partitions, aligning with the practical needs of this paper. Therefore, KL-divergence is selected as the metric.

$$S(\theta_2||\theta_1) = \sum_{i=1}^{|v|} p(s_i|\theta_2)\, log \frac{p(s_i|\theta_2)}{p(s_i|\theta_1)} \quad (5)$$

Here, θ represents the POI category with the highest intensity in the corresponding time segment, and s_i represents different

POI categories within the time segment.

The specific process mainly consists of two steps. Firstly, with 1 hour as the minimum granularity for time segments, the frequency of residents' arrivals at each POI category within each hour is counted. Then, the intensity of each POI category is calculated and sorted, with the POI category of the highest intensity being designated as the theme θ. Secondly, KL-divergence is used to measure the distance between adjacent time segments, and adjacent segments with a distance less than a certain threshold are merged, ultimately completing the division of time segments.

After dividing the time periods, the method for constructing activity graphs is applied to data from different time periods to obtain a set of community life circle activity graphs G^k for multiple time periods.

C. Representation Learning for Quantifying the Structure of Community Life Circles

After constructing multi-time-period community life circle activity graphs, collective embedding representation learning can be performed on these graphs to obtain a structural characteristic matrix that reflects the daily living patterns of residents within the community life circle.

During the encoding process, the community life circle activity graph is represented in a vectorized form using a dependency index. To ensure that each community has a unique feature vector after representation, embedding representation learning is conducted with the vectorized individual community life circle activity graph $G_t^{(k)}$ as the unit. Meanwhile, the vectorized community life circle activity graph $G_t^{(k)}$ contains eight different categories of POIs. To ensure that each POI category also has a unique feature vector after representation, facilitating comparisons between categories, the quantitative data of the eight POI categories in $G_t^{(k)}$ are converted into vectors and input into the autoencoder separately. Here, $\{y_{s,t}^{(k),1}, y_{s,t}^{(k),2}, \dots, y_{s,t}^{(k),m}\}$ represents the latent features in each hidden layer during the encoding step. $p_{s,t}^{(k)}$ is the represented feature vector for the eight POI categories and collectively forms the intermediate layer, i.e., the representation vector $p_t^{(k)}$ of the community life circle structure. The relationship between vectors at different levels is shown in Equation 6.

$$\begin{cases} y_{s,t}^{(k),1} = \sigma(w_{s,t}^{(k),1} g_{s,t}^{(k)} + b_{s,t}^{(k),1}), \forall t \in (1,2,\dots,n), \\ y_{s,t}^{(k),r} = \sigma(w_{s,t}^{(k),r} y_{s,t}^{(k),r-1} + b_{s,t}^{(k),r}), \forall r \in (2,3,\dots,m), \\ p_{s,t}^{(k)} = \sigma(w_{s,t}^{(k),m+1} y_{s,t}^{(k),m} + b_{s,t}^{(k),m+1}). \end{cases} \quad (6)$$

In the decoding step, the input is the latent feature vector $p_{s,t}^{(k)}$, and the final output is the reconstructed vector $\hat{g}_{s,t}^{(k)}$. $\{y_{s,t}^{(k),m}, y_{s,t}^{(k),m-1}, \dots, y_{s,t}^{(k),1}\}$ represents the latent feature vectors of each hidden layer during the decoding step. The relationship between vectors at different levels is shown in Equation 7, where w_s and b_s are the weight and bias terms, respectively.

$$\begin{cases} \hat{y}_{s,t}^{(k),m} = \sigma(\hat{w}_{s,t}^{(k),m+1} p_{s,t}^{(k)} + \hat{b}_{s,t}^{(k),m+1}), \forall t \in (1,2,\dots,n), \\ \hat{y}_{s,t}^{(k),r} = \sigma(\hat{w}_{s,t}^{(k),r} \hat{y}_{s,t}^{(k),r-1} + \hat{b}_{s,t}^{(k),r}), \forall r \in (2,3,\dots,m), \\ \hat{g}_{s,t}^{(k)} = \sigma(\hat{w}_{s,t}^{(k),1} \hat{y}_{s,t}^{(k),1} + \hat{b}_{s,t}^{(k),1}). \end{cases} \quad (7)$$

V. Experimental Analysis

This section conducts experimental evaluations on the proposed quantitative representation learning method for community life circle structures.

A. Dataset

The datasets used in this paper for experimental analysis are all real datasets. The residential community dataset is obtained from the real estate online website www.soufun.com, including a total of more than 13,000 residential communities in Beijing. The data comprises information such as names, latitudes and longitudes, addresses, and tags. The public service facility dataset is downloaded by calling the Baidu Map API interface, including POI (Point of Interest) data and AOI (Area of Interest) data, with a total of 1,871,019 POIs covering 8 major categories and 22 sub-categories. POI data includes information such as names, latitudes and longitudes, addresses, and tags; AOI data includes information such as names, addresses, areas, and latitude and longitude coordinates of boundary points. The unique identification code uid in the Baidu Map POI system is used to deduplicate the dataset, which achieves higher accuracy compared to other deduplication methods. Review data is obtained from the www.dianping.com website, containing information such as the names, addresses, prices, popularity, and reviews of POIs in various categories. Trajectory data includes Baidu Map route query trajectories, taxi trajectories, and shared bicycle trajectory data. Among them, the Baidu Map route query trajectory data comes from the Baidu Brain transportation dataset, and the shared bicycle trajectory data comes from the data competition community www.biendata.xyz, totaling 3,214,096 bicycle trajectory data points.

B. Evaluation Analysis

This paper evaluates the effectiveness of the proposed quantitative representation learning method for community life circle structures by predicting the convenience of these circles. The convenience of a community life circle refers to

the satisfaction level of community residents towards the public service facilities surrounding them.

A survey conducted in reference [28] investigated the living convenience of various regions in Beijing through questionnaires. The survey results classified the convenience of each community into five levels from 1 to 5. This paper adopts these survey results as the true convenience level labels for each community. Both existing representation methods and the proposed representation method in this paper are used to quantitatively represent the structure of community life circles, generating representation vectors. These vectors are then used as predictive attributes to predict the convenience levels of community life circles, and the different representation methods are evaluated based on the prediction results.

The specific representation methods are as follows:

(1) Explict Feature of Fixed Range (E-Fix): This method divides the life circle based on a fixed distance and quantifies the number of each type of POI within the life circle as features.

(2) Weekly Explict Feature of Fixed Range (WE-Fix): Reference [8] utilizes trajectory data between community residents and POIs, with a weekly cycle. Within a fixed-distance life circle, the number of each type of POI associated with residents on each day is quantified as a feature.

(3) Daily Explict Feature of Fixed Range (DE-Fix): Reference [27] employs a sliding window with KL-divergence as the metric to divide the time periods within each day for community residents. Using this as a cycle, within a fixed-distance life circle, the number of each type of POI associated with residents during each time period is quantified as a feature.

(4) Daily Latent Feature of Fixed Range (DL-Fix): This method divides time periods in the same way as DE-Fix. Utilizing the strategy proposed in Section 3.2.1, within a fixed-distance life circle, the quality of each type of POI associated with residents during each time period is quantified as a feature.

(5) Daily Explict and Latent Feature of Fixed Range (DEL-Fix): This method combines DL-Fix and E-Fix.

(6) Daily Explict and Latent Feature of kNN (DEL-kNN): Utilizing the strategy proposed in Section 3.2, this method employs the kNN approach to determine the necessary POIs within the life circle. It divides the day into multiple time periods using a sliding window and quantifies both the quality and quantity of each type of POI associated with residents during each time period as features.

Among them, methods (1), (2), and (3) are existing representation methods, while methods (4), (5), and (6) are the proposed representation methods in this paper. This paper selects five existing learning-to-rank algorithms to compare and evaluate the performance of the above six representation methods.

The five ranking algorithms include: (1) Multiple Additive Regression Trees (MART), an enhanced tree model that uses regression trees to perform gradient descent in function space; (2) RankBoost (RB), an enhanced pairwise ranking method that formalizes learning to rank as a binary classification problem for instance pairs, training multiple weak rankers and using their outputs as the final ranking result; (3) ListNet (LN), a list ranking method that employs neural networks and gradient descent as the model and algorithm; (4) RankNet (RN), which uses neural networks to model the underlying probability cost function; and (5) LambdaMART (LM), a combination of LambdaRank and MART, an enhanced version of MART based on RankNet and LambdaRank.

The ranking results are measured using Normalized Discounted Cumulative Gain (NDCG@N). A higher NDCG@N value indicates higher accuracy in the ranking evaluation, and vice versa.

The ranking results are shown in Table 2. From the table, it can be seen that the E-Fix method performs the worst, as it only counts the number of POIs and lacks sufficient acquisition of other features of the community life circle. The DE-Fix method outperforms the WE-Fix method because DE-Fix divides time periods into finer granularity, allowing for the acquisition of more dynamic information that aligns with the actual living situations of residents within the life circle.

The DL-Fix method is superior to the DE-Fix method because when establishing the connection between the community and POIs, DL-Fix considers the impact of POI quality on residents' activities, obtaining deeper information about residents' activities. Additionally, the DL-Fix method considers the issue of varying influence when the same POI is shared by multiple communities, allowing for a more reasonable allocation of POI influence.

The DEL-Fix method is superior to the DL-Fix method because DEL-Fix adds the E-Fix method on top of DL-Fix, combining dynamic and static, as well as explicit and implicit feature information, providing a more comprehensive and accurate description of the structural characteristics of the community life circle. The DEL-kNN method performs the best, mainly because it uses the kNN method to determine the necessary POIs within the life circle, removing redundant

POIs that have weak connections with life circle residents, which aligns more closely with the actual living situations of these residents.

Through experiments, it can be found that the proposed learning framework can effectively obtain dynamic and implicit features, making the evaluation of community life circle convenience more accurate.

Table 2 Comparison of convenience of community life circle

NDCG	Ranking methods	E-Fix	WE-Fix	DE-Fix	DL-Fix	DEL-Fix	**DEL-*k*NN**
@5	MART	0.468	0.405	0.438	0.462	0.525	**0.574**
	RN	0.405	0.3	0.435	0.441	0.474	**0.485**
	RB	0.354	0.216	0.345	0.363	0.504	**0.524**
	LM	0.438	0.462	0.375	0.465	0.573	**0.583**
	LN	0.414	0.3	0.414	0.441	0.47	**0.481**
@10	MART	0.471	0.411	0.459	0.48	0.594	**0.674**
	RN	0.456	0.3	0.486	0.492	0.516	**0.535**
	RB	0.375	0.384	0.381	0.39	0.546	**0.583**
	LM	0.483	0.477	0.42	0.528	0.564	**0.585**
	LN	0.477	0.327	0.498	0.492	0.512	**0.516**
@15	MART	0.489	0.42	0.462	0.507	0.621	**0.677**
	RN	0.495	0.336	0.537	0.546	0.558	**0.563**
	RB	0.39	0.462	0.375	0.411	0.612	**0.628**
	LM	0.471	0.522	0.522	0.543	0.636	**0.648**
	LN	0.516	0.336	0.537	0.546	0.554	**0.563**
@20	MART	0.495	0.489	0.471	0.525	0.642	**0.695**
	RN	0.516	0.366	0.588	0.594	0.6	**0.609**
	RB	0.441	0.471	0.474	0.474	0.627	**0.632**
	LM	0.564	0.582	0.555	0.624	0.666	**0.673**
	LN	0.576	0.366	0.579	0.594	0.596	**0.613**

In this section, the method proposed in this paper for identifying necessary POIs within a community life circle is compared with existing methods through experimental processes to demonstrate its effectiveness.

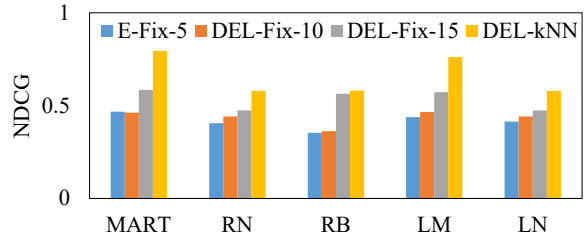
The specific comparison methods include the 5-minute, 10-minute, and 15-minute community life circles. The 5-minute community life circle employs the Explicit Feature of Fixed Range (E-Fix) representation method, while the 10-minute and 15-minute community life circles utilize the Daily Explicit and Latent Feature of Fixed Range (DEL-Fix) representation method, which combines explicit and latent features over a fixed range within a daily cycle. These are compared with the representation method proposed in this paper, which combines daily explicit and latent features using k-nearest neighbors (DEL-kNN).

The prediction results are shown in Figures 5(a)-(d), which compare five ranking learning methods with four representation methods for different life circle ranges. Observing the experimental results, it is evident that the prediction of convenience for the 5-minute life circle yields the worst performance. This is mainly due to the fact that walking is the primary mode of transportation for community residents within this range. Although pedestrian trajectory information can be obtained through GPS, mobile phone signaling, etc., it cannot be collected due to privacy protection concerns. Therefore, this paper adopts a static feature representation method. Additionally, since the 5-minute life circle covers a relatively small area, the POIs distributed within this range exhibit poor regularity, and the POI distributions vary significantly among different life circles, resulting in a weak correlation between life circle convenience and POI distribution. Consequently, none of the algorithms achieve satisfactory convenience prediction results for the 5-minute life circle. The convenience prediction for the 15-minute life circle is generally slightly better than that for the 10-minute life circle, primarily because the 15-minute life circle spatially aligns more closely with the actual activity range of community residents, allowing for access to more public service facilities necessary for daily life.

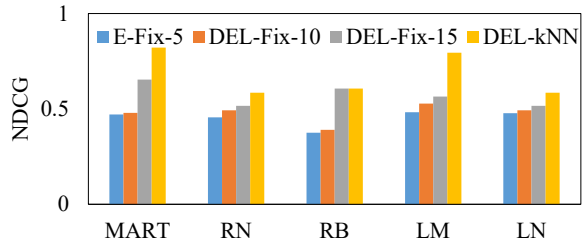
The method proposed in this paper for identifying necessary POIs within a life circle generally outperforms the 15-minute life circle in terms of convenience prediction. This is mainly because the proposed method not only removes redundant POIs that have weak connections with residents within the life circle but also considers the allocation of POI influence when shared by multiple communities. It can accurately identify POIs that have real and close connections with the community, leading to more accurate predictions of life circle convenience.

By comparing with existing POI identification methods within fixed-duration and fixed-distance life circles, the method proposed in this paper for identifying necessary POIs within a life circle aligns more closely with the actual living conditions of community residents and provides a new approach for delineating community life circles.

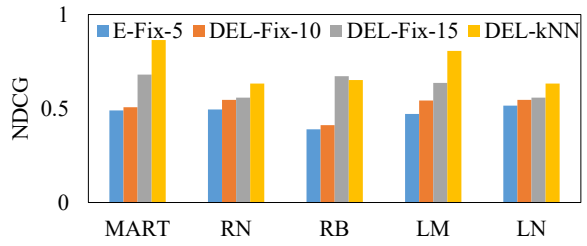
In this section, we compare different values of k in the k-nearest neighbors (kNN) algorithm to verify the validity of the k value selected in this paper. Specifically, k/2 and 2k are chosen as the comparison values, using the representation method that combines daily explicit and latent features based on kNN (DEL-kNN).



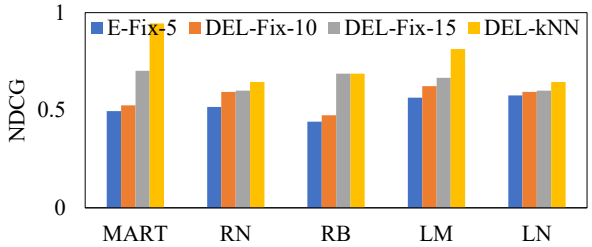
(a) NDCG@5



(b) NDCG@10



(c) NDCG@15



(d) NDCG@20

Fig. 5 Comparison of convenience in different ranges of community living circles

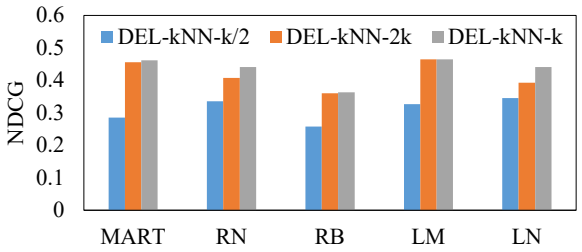
To analyze the relationship between the number of different types of POIs and the convenience of a community life circle, experiments are conducted with varying numbers of POIs. The selection of k for each type of POI draws on the conclusions from literature [24] regarding the k-nearest neighbor values for various facilities within a community life circle. The specific values are shown in Table 1.

Regarding the parameter selection for the autoencoder, although the choice of batch size is related to the number of samples in the training set, GPUs tend to perform better with batch sizes that are powers of 2. Therefore, setting batch sizes to 16, 32, 64, 128, etc., often results in better performance than setting them to multiples of 10. Experimental validation shows that the best experimental results are obtained when the batch size is 64. At this point, the corresponding number of layers in the autoencoder is 3, and the number of training epochs for the autoencoder is 20.

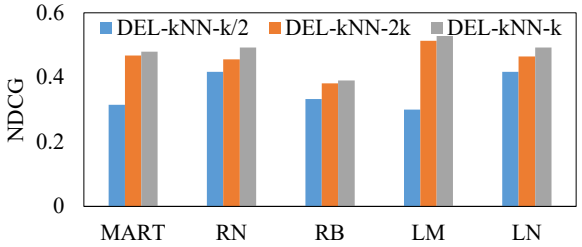
The prediction results are shown in Figures 6(a)-(d), which compare five ranking learning algorithms combined with three representation methods using different k values.

Observing the experimental results, we find that the prediction results for the convenience of the community life circle are the worst when k is set to k/2. This is because the number of POIs fails to meet the actual daily needs of community residents. The prediction results for k are generally slightly better than those for 2k. The main reason is that the number of POIs within the community life circle when k is used meets the actual living needs of community residents. Increasing the number of POIs leads to an increase in the number of useless POIs, which affects the accuracy of obtaining the relationship between useful POIs and community residents.

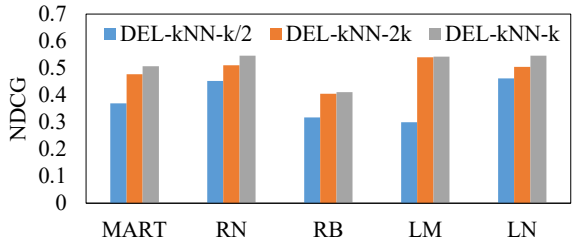
To better analyze the changes in personnel mobility between weekdays and holidays, this paper predicts the convenience of community life circles separately for weekdays and holidays, using the same comparison metrics and ranking learning algorithms as previously described.



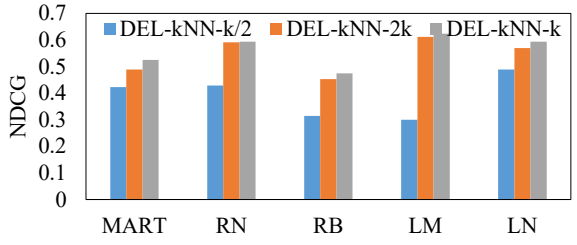
(a) NDCG@5



(b) NDCG@10

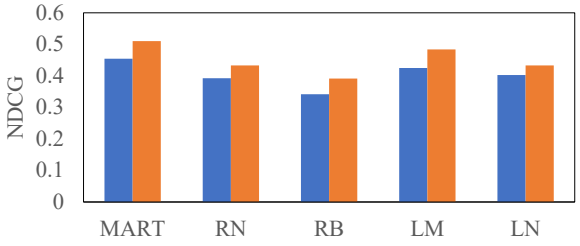


(c) NDCG@15

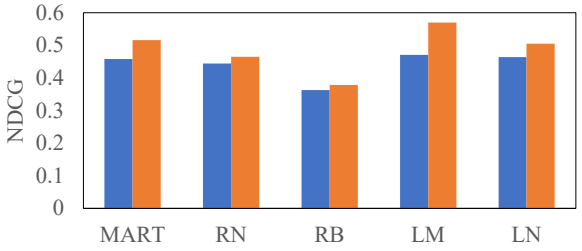


(d) NDCG@20

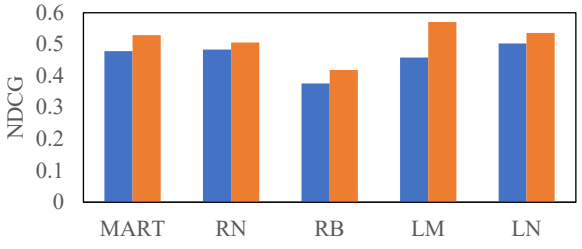
Fig.6 Comparison of convenience of community life circle with different *k* values



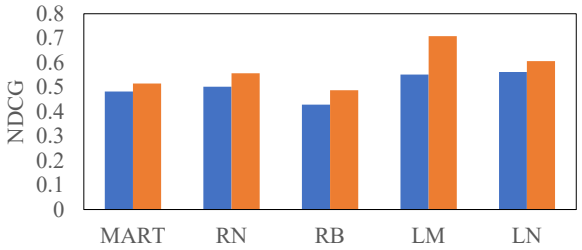
(a) NDCG@5



(b) NDCG@10



(c) NDCG@15



(d) NDCG@20

Fig.7 Comparison of convenience of community life circle between working days and holidays

Figure 7 presents a comparison of the results obtained using five ranking learning methods and two different time period representation methods. Figures 7(a)-(d) show comparisons of the top 5-20 returned results, respectively. From the results, it can be observed that the accuracy of convenience predictions for holidays is generally better than that for weekdays. The reason for this is that during weekdays, residents' main activity locations are concentrated near bus stops and subway stations, with a single category of POIs being visited. However, on weekends, residents tend to engage in recreational activities around their community life circles, with more obvious patterns. Therefore, the features of the community life circle obtained during holidays can better describe the structure of the community life circle, leading to more accurate predictions of convenience results.

VI. Conclusion

This paper studies the quantitative representation method for the structure of community life circles. The k-nearest neighbors (kNN) method is adopted to determine candidate Points of Interest (POIs) within the community life circle. By utilizing POI review data, scale data, and category weights, the dependency index of community residents on POIs is calculated, thereby establishing a connection between the community and POIs. The time periods of the activity diagram are divided by analyzing trajectory data reflecting residents' activity patterns. Collective embedding representation learning is performed on the activity diagram to obtain feature vectors that can stably represent the structure of the community life circle. For applications such as predicting the convenience of community life circles, experimental evaluations are conducted using real datasets, verifying the effectiveness of the proposed method.

ACKNOWLEDGMENT

This work is supported in part by the Project of the Educational Department of Liaoning Province (JYTMS20231596).

REFERENCES

[1] Sheng Yiyi. Research on the Division of Life Circles and Life Atmosphere in Central Urban Areas of Shanghai [C]. China Architecture & Building Press, 2019: 402-412.

[2] Qiang Siwei, Chen Xiaming, Jiang Kaida, et al. Analysis of Urban Spatio-Temporal Behavior Based on Mobile Network Traffic Logs [J]. Journal of Computer Research and Development, 2016, 53(4): 9. DOI: 10.7544/issn1000-1239.2016.20148278.

[3] Fan Jun, Tang Haoming, Ye Yu. Measurement of Community Life Convenience at the Human-Scale: A Refined Assessment Based on Multi-Source Urban Data [J]. New Architecture, 2020(05): 10-15.

[4] Shen Yuhui, Tong Ziyu. Research on the Evaluation Method of Community Life Circle Convenience at the Human-Scale [J]. South Architecture, 2022(07): 72-80.

[5] Sun Daosheng, Chai Yanwei, Zhang Yan. Definition and Measurement of Community Life Circles: A Case Study of Qinghe Area in Beijing [J]. Urban Development Studies, 2016, 23(09): 1-9.

[6] Wei Wei, Hong Mengyao, Xie Bo. Delimitation and Spatial Optimization of 15-Minute Life Circles in Wuhan Based on Supply-Demand Matching [J]. Planners, 2019, 35(04): 11-17.

[7] Zhang Bo, Zhao Yanyun, Zhou Fang. Spatial Clustering Research on the "15-Minute Community Life Circle" of Residential Areas - Based on POI Data [J]. The Survey World, 2019(01): 49-56.

[8] Wang P, Fu Y, Zhang J, et al. Learning Urban Community Structures: A Collective Embedding Perspective with Periodic Spatial-temporal Mobility Graphs [J]. ACM Transactions on Intelligent Systems, 2018, 9(6): 63.1-63.28.

[9] Chao Z, Zhang K, Quan Y, et al. Regions, Periods, Activities: Uncovering Urban Dynamics via Cross-Modal Representation Learning [C]// The 26th International Conference. International World Wide Web Conferences Steering Committee, 2017: 361-370.

[10] Fu Y, Liu G, Papadimitriou S, et al. Real Estate Ranking via Mixed Land-Use Latent Models [C]. ACM International Conference on Knowledge Discovery and Data Mining. 2015: 299-308.

[11] Laraqui M, Saaidi A, Mouhib A, et al. Images Matching Using Voronoi Regions Propagation [J]. 3D Research, 2015, 6(3): 27.

[12] Liu K, Yin L, Lu F, et al. Visualizing and exploring POI configurations of urban regions on POI-type semantic space [J]. Cities, 2020, 99: 102610.

[13] Gao Wei, Chen Xiangqi, Chen Sha. Typical Delimitation Methods and Applications of the 15-Minute Community Life Circle - A Case Study of Jinniu District, Chengdu [J]. Habitat, 2022(02): 13-19.

[14] Chen Lufeng, Li Chengming, Dai Zhaoxin, et al. Research on the Configuration of Public Service Facilities in the "15-Minute Life Circle" of Residential Areas [J]. Surveying and Mapping Science, 2022, 47(01): 236-244.

[15] Tan P Y, Samsudin R. Effects of spatial scale on assessment of spatial equity of urban park provision [J]. Landscape & Urban Planning, 2017, 158: 139-154.

[16] Huang Huiming, Zhou Dailin, Wang Ye. Research on the Spatial Organization Mode of Community Life Circles Based on Residential Morphological Types - A Case Study of Guangzhou [J]. Urban Planning Forum, 2021(02): 94-101. DOI: 10.16361/j.upf.202102015.

[17] Yu Zheng, Shangguan Zongshan, Zhou Dandan, et al. Optimal Allocation of Public Service Facilities in Future Community Life Circles Based on Spatio-Temporal Impact Factors [J]. Surveying and Mapping Bulletin, 2022(08): 143-148+154.

[18] Chai Y, Chunjiang LI, Xia W, et al. Study on the Delineation Model of Urban Community Life Circle: Based on Qinghe District in Haidian District, Beijing [J]. Urban Development Studies, 2019.

[19] Liu X, Long Y, et al. Automated identification and characterization of parcels with OpenStreetMap and points of interest [J]. Environment & Planning B Planning & Design, 2015, 43(2): 341-360.

[20] Zhang Y, Licia C, et al. Urban computing: concepts, methodologies, and applications [J]. ACM transactions on Intelligent Systems and Technology, 2014, 5(3): 38.1-38.55.

[21] Ou M, Cui P, Pei J, et al. Asymmetric transitivity preserving graph embedding [C]. ACM International Conference on Knowledge Discovery and Data Mining. 2016: 1105-1114.

[22] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798-1828.

[Note: Reference [23] is a duplicate of [21] and has been omitted.]

[24] Luo Xueyao, Zhang Wenjia, Chai Yanwei. Research on the Threshold Effect of the Built Environment in the 15-Minute Life Circle [J]. Geographical Research, 2022, 41(8): 2155-2170.

[25] Erath A, Maheshwari T, Joos M, et al. Visualizing Transport Futures: The Potential of Integrating Procedural 3D Modelling and Traffic Micro-Simulation in Virtual Reality Applications [C]. Transportation Research Board Annual Meeting. 2017.

[26] Fu Y, Hui X, Yong G, et al. Exploiting Geographic Dependencies for Real Estate Appraisal: A Mutual Perspective of Ranking and Clustering [C]// ACM International Conference on Knowledge Discovery and Data Mining. 2014: 1047-1056.

[27] Sun Huanliang, Peng Cheng, Liu Junling, et al. Representation Learning for "15-Minute Life Circle" Community Structures [J]. Journal of Computer Applications, 2022, 42(06): 1782-1788.

[28] X Zhang, Du S, Zhang J. How do people understand convenience-of-living in cities? A multiscale geographic investigation in Beijing [J]. ISPRS Journal of Photogrammetry & Remote Sensing, 2019, 148(FEB.): 87-102.

Zheng Shiyu (2000-), female, master. Her research focuses on artificial intelligence.

Liu Junling(1972-), female, associate professor. Her research focuses on artificial intelligence and data mining.

Traffic Route Knowledge Extraction Model from Social Media Data

You HangYu Sun Huanliang
(*School of Computer Science and Engineering, Shenyang Jianzhu University, Shenyang 110168, China*)

Abstract—Using social media data for traffic strategy in route planning offers better interpretability and more closely aligns with user preferences compared to electronic navigation systems. However, unstructured traffic strategy texts often contain redundant information and inconsistencies, making them difficult to directly apply to route recommendation. By automatically extracting traffic route information to build a knowledge graph, this valuable data can be better utilized for route planning. Due to the presence of domain-specific entities and attributes in traffic strategy texts, existing methods face limitations when dealing with such long texts and require a large amount of manually labeled data. Moreover, current research often overlooks users' textual descriptions of routes, which results in the failure to effectively leverage key information such as route experiences and evaluations. Thus, large language models can be used to extract traffic route information from traffic strategy texts and build a traffic travel event knowledge system. By combining the knowledge system with prompt templates for traffic event extraction, large models' few-shot learning capabilities can be employed for data labeling, creating a dataset for instruction fine-tuning. This leads to the fine-tuning of a smaller language model using the LLaMA-Factory framework to develop a traffic travel event extraction model, TRKE, for extracting traffic travel events from traffic strategy texts. Finally, a structured traffic route knowledge graph is constructed using the extracted information. Experimental results show that the model achieved promising results in traffic event extraction, with an accuracy rate of 62.02%.

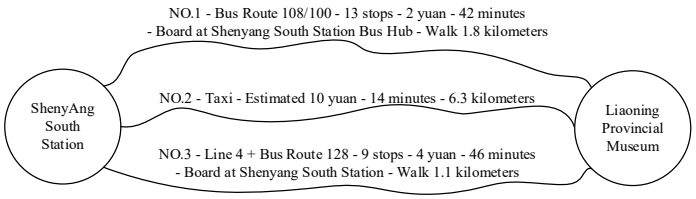
Keywords—event extraction, knowledge extraction, Large Language Model, traffic route planning

I. INTRODUCTION

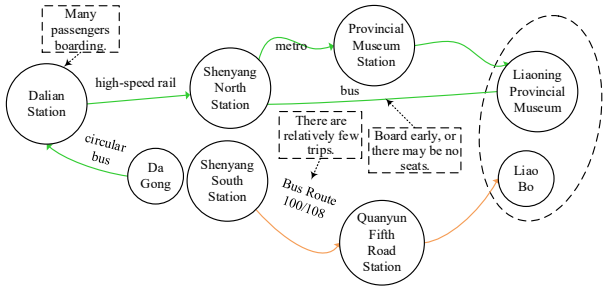
In daily travel, electronic map platforms are typically used for route planning, which requires the use of road network data and users' historical trajectory data for real-time route planning. This is achieved through traffic prediction [1] and historical trajectory analysis [2], using factors such as route attributes like distance, travel time, and the number of transfers.

To enhance the experience of traffic route planning, many scholars have proposed various traffic route query and recommendation solutions. Early research often used shortest path query algorithms based on specific criteria (such as shortest time, lowest cost, or shortest walking distance) to perform route queries [3]. With the advancement of machine learning and deep learning models, many studies have analyzed large-scale historical travel trajectories, revealing relationships between users, travel routes, and transportation modes, thus providing more refined traffic route recommendations [4].

In addition, a large amount of traffic guide text from social media plays an important role in route recommendations. These traffic guides not only include textual descriptions of routes but also contain real-time reviews and experience information. If subjective information from these guides could be integrated into navigation routes, they would provide more comprehensive data for route selection. Moreover, electronic map platforms still face some limitations in small towns and remote areas, where location and route information is often incomplete or outdated. Us-er-contributed traffic guides can effectively supplement geo-graphic and route information in these regions, helping to fill in these gaps.



(a) The existing navigation system returns the route



(b) A route example with user experience

Fig.1 Traffic Route Knowledge Graph

Fig. 1(a) shows a public transportation route from "Shenyang South Station" to the "Liaoning Provincial Museum" obtained from an existing electronic navigation system, including information such as travel time, distance, and cost. This information helps users understand the road conditions in advance and plan their trips more reasonably. However, there is also a related traffic guide on social media: "Got off at Shenyang South Station and headed straight to the Liaoning Provincial Museum. The 100 and 108 buses recommended by the map never came, so I took a taxi and wasted half an hour waiting." This information indicates that the traffic route recommendation from the electronic map may sometimes not match reality. Therefore, extracting and attaching user experience information to navigation routes can effectively enhance users' understanding of the route and provide more thorough guidance for route selection.

Fig. 1(b) shows a route example with user experiences, with the experiences extracted from Table I's traffic guide from "Dalian" to "Shenyang." As seen in the figure, this route map not only includes basic information provided by the existing navigation system but also incorporates user experience information, such as feedback on the traffic lines "There are relatively few trips." and suggestions like "Board early, or there may be no seats." There is also an evaluation of the route node, such as "Many passengers boarding at Dalian Station."

Table 1 Example of Information Extraction for Traffic and Travel Events

Enter text: Starting from Dalian to Shenyang! When you come to Shenyang, you must visit the LiaoBo. We took the reserve train from DaGong to Dalian Station, then took the high-speed rail to Shenyang North Station. Dalian Station is the starting station, so there are many passengers getting on. After exiting the station, take the subway heading towards Taoxian Airport and get off at Provincial Museum Station, and you'll be right there. There is a shuttle bus back to Shenyang North Station in front of the museum. Get on early, or there may be no seats left. Nearby food recommendation: I suggest "Qianlima Clay Stove Barbecue" on South Fourth Jing Street. It's tasty and affordable. Friendly reminder: No flash photography, no eating while viewing the exhibits, and no running or jumping inside the museum.		
no-prompt template (semi-structured form with natural language characteristics.)	with prompt template && without instruction fine-tuning (structured form)	with prompt template && instruction fine-tuning (correctly extracted structured form).
1. Departure location: Dalian, specifically starting from DaGong. **2. Departure method**: Take the reserve train to Dalian Station. **3. High-speed rail information**: Take the high-speed train from Dalian Station to Shenyang. There is a large flow of passengers boarding at Dalian Station. **4. Arrival station**: The destination is Shenyang North Station. **5. Shenyang city transportation**: After exiting the station, transfer to the subway line heading towards Taoxian Airport. Get off at Provincial Museum Station, which is located at the Liaoning Provincial Museum. **6. Returning to Shenyang North**	**Event 1:** "**Departure location**": "Dalian" "**Destination":** "Shenyang" "**Starting station**": "Dalian Station" "**Arrival station**": "Shenyang North Station" "**Transportation mode**": "High-speed rail" "**Notes and suggestions**": "There are many passengers boarding at Dalian Station" **Event 2:** "**Departure location**": "Shenyang North Station" "**Destination**": "Taoxian Airport" "**Arrival station**": "Provincial Museum	**Event 1:** "**Departure location**": "DaGong" "**Destination**": "Dalian Station" "**Transportation mode**": "Reserve train" **Event 2:** "**Departure location**": "Dalian Station" "**Destination**": "Shenyang North Station" "**Transportation mode**": "High-speed rail" "**Notes and suggestions**": "There are many passengers boarding at Dalian Station" **Event 3:** "**Departure location**": "Shenyang North

Station: There is a direct bus available at the museum entrance, which you can take back to Shenyang North Station. It is recommended to board early to secure a seat. **7. Nearby food recommendation**: Qianlima Clay Stove Barbecue: Located on South Fourth Jing Street. **8. Friendly reminders**: Do not use flash when taking pictures. No eating in the exhibition area. Avoid running or jumping.	Station" "**Transportation mode**": "Subway" **Event 3:** "**Departure location**": "Museum entrance" "**Destination**": "Shenyang North Station" "**Transportation mode**": "Bus" "**Notes and suggestions**": "Board early, or there may be no seats"	Station" "**Destination**": "Liaoning Provincial Museum" "**Arrival station**": "Provincial Museum Station" "**Transportation mode**": "Subway" **Event 4:** "**Departure location**": "Liaoning Provincial Museum entrance" "**Destination**": "Shenyang North Station" "**Transportation mode**": "Bus" "**Notes and suggestions**": "Board early, or there may be no seats"

Since user-shared traffic guides are usually presented in unstructured text and users have different ways of expressing themselves, extracting traffic information is challenging. These guides often consist of multiple traffic events, each representing part of the route. Although using large models for information extraction is a relatively advanced technique [5], applying them directly to traffic event extraction still presents many difficulties.

First, the diversity of user-shared strategies may cause large models to overgeneralize, extracting irrelevant content. Second, due to the lack of specialized training on specific geographical vocabulary and transportation knowledge, large models may struggle to extract uncommon terms, even with extraction templates. Additionally, traffic information contains complex entities and relationships, such as long-distance entities, referential entities, and distracting elements, which increase the complexity of extraction.

To address these challenges, a traffic event extraction template was developed using prompt engineering, guiding the large model to focus on key traffic-related elements such as departure point, destination, mode of transportation, and time. This method enables the model to better understand the task requirements, recognize the relevant context, and improve extraction accuracy. Furthermore, instruction fine-tuning was employed to optimize the large model's performance in traffic event extraction tasks. By building a dataset using the traffic event extraction template and using LoRA to fine-tune the Qwe1.5-14B-Chat model, the model gained a deeper understanding of transportation-specific knowledge and vocabulary, enhancing its contextual comprehension and reasoning abilities, thereby improving the accuracy and comprehensiveness of the extracted information.

The main contributions of this paper include:

1) Constructing a traffic event knowledge system and designing a traffic event extraction template to guide the extraction of traffic guide information.

2) Designing a modular prompt construction strategy to activate the large model's information extraction capabilities and guide it in constructing a traffic event dataset through few-shot learning.

3) Proposing a large model-based traffic information extraction method, using the traffic event dataset to fine-tune the model with instructions, improving the accuracy of traffic route knowledge extraction.

4) Conducting extensive experiments to validate the designed traffic route knowledge extraction model, demonstrating the effectiveness of the proposed method. A traffic route map with user experience knowledge was constructed, making route recommendations more in line with actual needs.

II. Related Work

This section introduces the related work in three aspects: traffic route query, traffic route recommendation, and information extraction.

A. Traffic Route Query

Traffic route query is an important application in location-based services and can be defined as the problem of finding an optimal path between a specified pair of start and end points within a given traffic network.

In daily travel scenarios, traffic networks often change based on travel time. Literature [6] proposes a highly balanced tree structure index, TD-G-tree, for time-dependent real road networks. Based on this index, an efficient algorithm that

com-bines dynamic programming and time partitioning techniques is designed to support time-dependent path planning. Literature [7] presents a two-stage route planning framework called MOOP, which is used to plan efficient routes within a time budget in a dual time-varying urban road network.

B. Traffic Route Recommendation

With the widespread use of location collection devices, large amounts of historical trajectory and traffic data have become easier to access. Using this historical trajectory data can reveal user habits and preferences, which play an important role in improving traffic route recommendations. Therefore, many scholars have studied the use of large amounts of trajectory data to mine or learn users' travel patterns, thereby improving route planning performance.

Literature [8] constructs a transition probability model based on segment selection within a road network from a large number of location trajectories, thus mining the most popular routes. Literature [9] proposes a path planning method called DRPK, which captures dependencies between road segments using historical trajectories and trains a classification model to detect key segments. The complete route is then divided into segments for query, significantly improving the efficiency and accuracy of path planning.

Additionally, discovering user preferences and travel pat-terns from historical trajectory data can also provide personalized route recommendations. Literature [10] mines users' personal travel preferences and behaviors from their historical GPS trajectories and uses collaborative filtering techniques to predict the frequency of their route choices. A Naive Bayes model is then used to generate personalized travel routes with the highest probability of user selection. Literature [11] enhances the A* search algorithm with neural networks, using contextual information to capture trajectory features, effectively providing users with more complex personalized route recommendations. Liter-ature [12] proposes a public transportation route engine based on users' historical travel data, which establishes an urban transportation model by combining user query data with multiple travel cost factors (such as time or distance), and uses route ranking to reflect user preferences.

This paper mines user travel route trajectories from social media text, recording users' route evaluations and experiences described in their language while capturing the travel trajectories. These textual descriptions can effectively enhance the interpretability of route recommendations.

C. Information Extraction

Information extraction refers to the automatic identification and extraction of structured information from unstructured text data. Common methods of information extraction include rule-based approaches [13], machine learning-based approaches [14-16], and deep learning-based approaches [17,18].

Rule-based approaches require analyzing domain-specific data and establishing corresponding rule templates. Information is extracted by pattern matching with the text. Literature [13] uses expert knowledge and experience to construct event extraction templates and continuously improves and updates these templates to meet the needs of different domain tasks. While rule-based information extraction methods can ensure accuracy in small-scale tasks, they rely on manually defined rule sets by experts, resulting in weak generalization capabilities and difficulty in scaling up for large tasks.

Machine learning-based approaches require fully or partially annotated training corpora to automatically learn features and patterns from the text. These methods have strong adaptability and generality, making them suitable for large-scale open-domain information extraction tasks. Literature [14] uses conditional random fields for sequence labeling, introduces feature induction mechanisms to select and generate features, and utilizes enhanced dictionary integration to significantly improve the performance of named entity recognition. Literature [15] improves the accuracy of relation classification by combining lexical, syntactic, and semantic knowledge, using support vector ma-chines as classifiers, and leveraging multi-feature combinations, demonstrating the importance of feature engineering in relation extraction. Literature [16] uses a maximum entropy classifier based on input text features to extract event triggers, arguments, and related elements, making classification decisions by maximizing information entropy. However, machine learning-based methods are constrained by feature selection and manually an-notated training sets, making it difficult to discover implicit or complex relationships from large-scale text, and they suffer from feature extraction error propagation.

Deep learning-based approaches utilize neural network models and have shown excellent performance in information extraction tasks. These methods can automatically extract high-level features from data, thereby improving the accuracy and generalization capabilities of information extraction. Liter-ature [17] proposes a multi-task information extraction frame-work that uses BERT to generate context-sensitive span representations, capturing rich contextual information in the

text. This approach enables handling entity recognition, relation extraction, and event detection tasks within the same framework. Literature [18] converts the event extraction task into a multi-round question-answering task, using a pre-trained language model-based QA system to gradually extract event triggers and arguments in a dialogue format. Although deep learning-based information extraction models can handle more complex tasks, they still require large amounts of data and computational resources, and their interpretability is relatively poor, making it difficult to directly integrate domain experts' knowledge and rules.

The emergence of large language models (LLMs), such as GPT [19], LLAMA [20], and Qwen [21], has advanced the field of information extraction, with many studies leveraging LLMs for information extraction [22]. The deep understanding capabilities, context awareness, few-shot learning, zero-shot learning, and generalization abilities of LLMs make them highly accurate in handling complex and implicit information extraction tasks. However, due to the lack of specialized training in specific do-mains, large models may encounter challenges when dealing with domain-specific terminology and knowledge. Instruction fi-ne-tuning methods are required to further train these models on specific tasks to improve their performance in specialized do-mains [23].

In this paper, prompt instruction descriptions and task examples are first constructed for the traffic event extraction task, utilizing the few-shot learning capabilities of large models for data annotation. Additionally, a smaller model is used for instruction fine-tuning to better adapt to specific use cases, thereby achieving accurate extraction of information related to traffic events.

III. METHODOLOGY

This paper proposes a large language model-enhanced traffic route knowledge extraction model, TRKE. After data acquisition and preprocessing, the traffic event extraction is completed through steps including constructing the knowledge system and extraction templates, generating a fine-tuned dataset, and building a fine-tuned extraction model. The detailed process is de-scribed in the following sections.

A. Data Acquisition and Preprocessing

This paper crawls and collects traffic-related texts published on social media platforms, such as travel notes, traffic announcements, and travel timetables. Since the collected information includes not only traffic guide texts that contain rich traffic events and route knowledge but also other irrelevant information, it is necessary to filter the collected texts through text classification. A classification example is shown in Table 2.

Table 2 Text Corpus Classification Example

Example	Category Label
Take the high-speed rail from Shenyang, and it takes about one hour to reach Fengcheng East Station.	Traffic Event Guide Text
Liaoning Art Museum has two floors, and many people take photos there. The staircase is a great spot for photos, and good lighting also helps.	Location Guide Text
Looking for a ride on February 2nd in the morning from Hangzhou to Anshan.	Non-Guide Text

First, all Points of Interest (POI) names under the "Nearby Attractions" category in various cities of Liaoning Province are crawled from the DazhongDianping platform. Then, using the keyword combination "POI name + traffic guide," related guides in Liaoning Province are searched on the Xiaohongshu platform. After identifying and eliminating duplicate texts, a total of 12,531 blog posts were collected.

In the text classification task, existing large models only need a few task samples to understand the task requirements and complete the classification task. Therefore, this paper directly uses a large model to classify the collected blog posts. However, after multiple experiments, it was found that directly classifying texts into "Traffic Event Guide Texts" and "Other Texts" using the large model did not yield satisfactory results. Hence, prompt engineering was employed to guide the large model in first classifying the input text as either "Guide Text" or "Non-Guide Text," followed by a second classification of "Guide Text" into "Traffic Event Guide Texts." The following text classification prompt template was ultimately generated. Using prompt engineering to guide the large model for text classification effectively improves the model's classification accuracy.

Text Classification Prompt Template:

```
{
  "instruction": "You are an expert in text classification. Classify the input content: First, determine if it is a guide-type text. If not, label it as 'Non-guide text'; otherwise, determine if this guide text contains a traffic event. If it does, label it as 'Traffic event guide text'; if not, label it as 'Location guide text'. Only output the label.",
  "schema": [
    "Non-guide text",
    "Traffic event guide text",
    "Location guide text"
```

```
  ],
  "examples": [
    {
      "input": "Example 1",
      "output": [
        "Answer1"
      ]
    }
  ],
  "input": "Traffic guide text input"
}
```


To test the effectiveness of large models in the classification task and select the most suitable large model for this text classification task, 500 manually labeled data points were used as a test set, and multiple experiments were conducted to compare several large models, including: abab6.5-chat, GLM-4, Baichuan4, moonshot-v1-8k, Qwen-max, and Yi-large. The experimental results show that the F_1 score of moonshot-v1-8k (88.44%) is higher than that of the Qwen-max model (84.95%), but the Precision score of Qwen-max (89.77%) is higher than that of moonshot-v1-8k (87.13%). Since this paper aims to obtain more accurate traffic guide text data, the Qwen-max model was chosen as the classification model for traffic guide texts.

Through the above data acquisition and preprocessing methods, a total of 6,466 "Traffic Event Guide Texts" were obtained, providing a foundation for the subsequent model training tasks.

B. Knowledge System and Extraction Template Construction

Traffic guide texts contain a wealth of traffic route knowledge, which is distributed across different traffic routes and presented in the form of traffic events within the text. To better utilize this knowledge, it is necessary to extract and organize these traffic events in a structured manner so that they can pro-vide valuable references for real-world travel. To achieve this goal, it is essential to establish a traffic event knowledge system and extraction templates to guide large models in knowledge extraction.

The traffic event knowledge system can effectively organize and represent traffic route knowledge, clearly identifying key entities, the relationships between them, and their attributes. The traffic event extraction template, based on the knowledge system, not only provides a structured framework for information ex-traction but also offers rich semantic information. This allows the extraction process to rely not only on lexical and syntactic analysis but also on semantic information to aid extraction, thereby improving the comprehensiveness and accuracy of the information extraction.

In this paper, a bottom-up approach from domain ontology construction is employed. By analyzing user-generated traffic guides, traffic-related entities are mined, classified, and generalized to generate a traffic event knowledge system.

First, the large model is used to extract all traffic-related entities, such as travel elements and experiences, from 1,293 traffic event guide texts as arguments in traffic events. For example, from the sentence "We arrived in Shenyang at ten o'clock," "Destination: Shenyang" and "Arrival time: ten o'clock" are extracted. Since the argument names generated by the large model vary, such as "End point" and "Destination," which have similar meanings, the large model is used to merge all arguments through prompts like "merge synonymous arguments" and "merge identical concept arguments." Based on the frequency of argument occurrences in the traffic guide texts, all merged arguments are filtered. Ultimately, twenty-nine types of event arguments were selected, including seventeen types of travel elements and twelve types of travel experiences, which together form the traffic event knowledge system. This knowledge system includes location, time, and mode of transportation, covering the factors that influence travel route planning and decision-making.

After constructing the traffic event knowledge system, corresponding relationships were added to all event arguments, forming the traffic route knowledge graph ontology structure, as shown in Fig. 2.

Traffic events based on the traffic knowledge system can provide users with interpretable decision-making references, such as why a recommended route is superior, thereby enhancing service credibility and improving user acceptance.

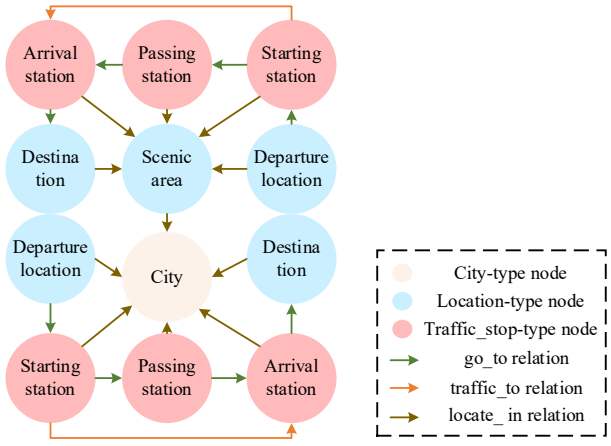


Fig. 2 Ontology Structure of the Traffic Route Knowledge Graph

Since event extraction tasks typically consist of two sub-tasks: event detection and event argument identification, this study constructs a traffic event extraction template that

extracts both triggers and arguments. As this study focuses on the field of traffic, the event types in the template are restricted to traffic events. In addition, based on the traffic event knowledge system constructed earlier, a corresponding set of event argument roles is formed. The following JSON-formatted traffic event extraction template is generated to guide the large model in traffic event extraction tasks:

```
"schema": [
  {
    "event_type": "traffic and transportation ",
    "trigger": true,
    "arguments": [
        "City", "Departure location", "Destination",
        "Starting station", "Passing station", "Arrival station",
        "Entrance gate", "Exit gate", "Mode of transportation",
        "Scenic area", "Route name", "Departure frequency",
        "Departure time", "Arrival time", "Duration",
        "Cost", "Ticket purchase location",
        "Travel experience", "Notes and suggestions",
        "Parking information", "Road condition information",
        "Highway information", "Route and navigation",
        "Operational information", "Departure time",
        "Ticket price and payment", "Station information",
        "Recommendations and reviews", "Other experiences"
    ]
  }
]
```

The traffic event extraction template constructed through the above method not only provides the large model with clear output format requirements but also integrates domain knowledge. By combining task templates with domain knowledge, the model can better overcome the shortcomings of lacking expertise in specific domains, thereby improving the accuracy and completeness of event extraction.

C. Fine-tuned Dataset Generation

To reduce the cost of manual annotation, this paper utilizes the few-shot learning capabilities of large models to generate a dataset for model fine-tuning. In large models, users interact through prompts in the form of instructions. For different types of tasks, task-specific prompts need to be designed to achieve the desired results. To better utilize the large model for text classification and information extraction tasks, a modular prompt strategy for vertical domain dataset construction is proposed. This strategy includes three modules: the task description module, task schema module, and task example module.

(1) Task Description

In prompt construction, the task description module is a key step to ensure the large model correctly understands and completes the assigned task. This paper uses a combination of role prompts and chain-of-thought reasoning strategies. First, the role the large model plays is clearly defined, such as "event extraction expert" or "text classification expert." This setting can effectively stimulate the large model's knowledge related to the specified task, helping it better understand and execute the task requirements. Additionally, the chain-of-thought reasoning strategy breaks down a complete task into multiple logical steps, guiding the large model to solve the problem according to the specified reasoning process. For example, in the text topic classification task, reasoning is completed through multiple judgment steps. This step-by-step reasoning pattern not only aligns with human cognitive processes but also provides the language model with clear decision logic. Finally, in the task description, the model's output format is restricted to a structured JSON string format.

(2) Task Schema

This module provides a basic framework structure for the large model to solve specific problems, allowing it to generate structured output according to the expected format. Particularly in the traffic event extraction task, a clear schema framework can provide effective guidance for the large model, helping it accurately identify and extract all details related to traffic events in complex input texts.

(3) Task Example Module

The task example module provides the large model with specific references and learning objects through high-quality example data. By learning the mapping relationship between input and expected output, the large model can better understand the task requirements and the abstract concepts in the task template. This example-based learning approach effectively activates the few-shot learning capabilities of the large model, allowing it to generalize to new inputs based on limited example samples. Additionally, the reasonable design of task examples can better cover different cases and edge conditions of the task.

Finally, the task description, task schema, task example, and input text are integrated into a JSON-formatted dictionary string as a prompt to interact with the large model, completing the assigned natural language processing task. Below is a sample prompt for the traffic event extraction task:

Traffic and Transportation Event Extraction Prompt Template

```
{
  "instruction": "You are an expert in event extraction. Please
extract events from the input that match the schema definition.
Return an empty list for non-existent events and 'NAN' for missing
```

arguments. If an argument has multiple values, please return them as a list. Answer in JSON string format. You can refer to the examples for extraction.",

```
"schema": [
  {
    "event_type": "Traffic and Transportation",
    "trigger": true,
    "arguments": [
      "City",
      "…",
      " Other experiences "
    ]
  }
],
"examples": [
  {
    "input": "Example 1",
    "output": [
      "Answer1"
    ]
  }
],
"input": "Traffic guide text input"
}
```

The large model will generate the corresponding output based on the complete prompt, serving as the final structured answer to the task. Additionally, to further enhance the large model's performance in information extraction tasks, an instruction fine-tuning dataset needs to be constructed. The format of this dataset follows the Stanford-developed Alpaca model [24], which contains three core fields: "instruction," "input," and "output." The "instruction" field is composed of the task description module, task framework/template module, and input text, and is represented as a JSON-formatted dictionary string. This dictionary contains detailed task descriptions, the expected output format, and comprehensive context information, providing the large model with clear task instructions. The structured answers generated by the large model based on the input text are manually corrected and used as the result in the "output" field. After leaving the "input" field empty, these three parts are combined to form the complete fine-tuning dataset. By constructing a fine-tuning dataset, the large model can better capture task characteristics, learn the output format, and thus improve its ability in information extraction tasks.

D. Fine-tuned Extraction Model Construction

To better align the large model's traffic event extraction capabilities with the requirements of constructing a traffic route knowledge graph, this paper uses the fine-tuned dataset to per-form instruction fine-tuning on the Qwen1.5-14B-Chat model. Instruction fine-tuning involves constructing a fine-tuned dataset containing task instructions, inputs, and expected outputs, and further fine-tuning the pre-trained model to enhance its ability to understand and execute specific task instructions, thus optimizing its performance in downstream tasks.

Due to the large number of parameters in the model and the relatively long input and output text length in this study, choosing a full-parameter fine-tuning approach would result in high training costs and catastrophic forgetting of the large model. Therefore, this study selects the Low-Rank Adaptation (LoRA) method [25], an efficient parameter fine-tuning approach, to perform instruction fine-tuning on the large model. LoRA adjusts the pre-trained model's parameters by inserting trainable small matrices instead of directly fine-tuning all the model's parameters.

As shown in Fig. 3, for the original parameter matrix $W_0 \in \mathbb{R}^{d\times k}$, in order to equip it with the ability to perform downstream tasks, it needs to be updated using $\Delta W \in \mathbb{R}^{d\times k}$. To improve computational efficiency, LoRA uses low-rank decomposition to break ΔW into two smaller matrices $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$, where $r \ll min(d, k)$.. During model training, W0 is frozen, and only the parameters A and B are updated. If the model input is x, the output h is as shown in equation (1).

$$h = W_0 x + \Delta W x = W_0 x + BAx \quad (1)$$

Moreover, the plug-in nature of LoRA allows the LoRA models trained on other datasets to be reused. To enhance the model's contextual understanding and reasoning abilities during information extraction, the LoRA model trained on the general information extraction dataset IEPile [26] is further trained on the traffic event extraction task, resulting in the final TRKE traffic route knowledge extraction model.

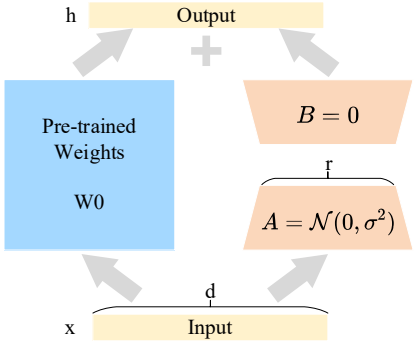


Fig. 3 LoRA Fine-Tuning Structure

IV. EMPERIMENTAL EVALUATIONS

A. Experimental Environment and Parameters

All experiments in this paper were conducted on a cloud

server platform, with the operating system being Ubuntu 20.04 and the development environment being PyTorch 2.1.0, running in Python 3.10. The inference and instruction fine-tuning environment consisted of 100GB of memory and an NVIDIA A800-80GB GPU.

During the inference phase of the large model, the decoding parameters determine the diversity of the generated text. To reduce the randomness of text generation, ensuring higher accuracy of the extracted information and consistency with the input text, the decoding parameters were set with a Temperature of 0.1 and Top-k of 1.

During the fine-tuning training phase, the Adam optimizer was used in combination with the cosine_with_restarts learning rate adjustment strategy to dynamically adapt the learning rate, and BF16 precision was used to improve computational efficiency and training speed. Since the LoRA method was adopted for fine-tuning, the settings of the LoRA parameters were crucial for model performance. After multiple parameter optimization experiments, it was found that when LoRA_r was set to 32 and LoRA_alpha to 64, the model performance was optimal. Other experimental parameter settings are shown in Table 3

Table 3 Instruction to Fine-Tune Model Parameters

Parameter Name	Parameter Value
Learning rate	5e-5
LR scheduler	cosine_with_restarts
Optimizer	Adam
Epochs	10
Batch size	4
Lora rank	32
Lora alpha	64
Lora dropout	0.1
Max length	3500

B. Evaluation Metrics

This study divides traffic event arguments into travel element arguments and travel experience arguments. For travel element arguments, a strict match method was used to calculate the number of correctly extracted arguments. Since the experiment uses a generative model for information extraction, handling traffic experience statements, which often include complex and diverse expressions, implicit semantic connections, and strong contextual dependencies, introduces some uncertainty during text extraction. Therefore, to evaluate the extraction effectiveness of traffic experience arguments, the ROUGE-2 metric was used to calculate the textual similarity between the extracted answers and the reference answers. Results exceeding a predefined similarity threshold were considered correct extractions. The ROUGE-2 calculation method is shown in Equation 2. After calculating the number of correctly extracted arguments, precision (P), recall (R), and F_1 score were used as the evaluation metrics for model performance.

$$\text{ROUGE-2} = \frac{\sum_{\text{gram}_2 \in \text{RefIE}} \text{Count}_{\text{match}}\left(\text{gram}_2\right)}{\sum_{\text{gram}_2 \in \text{RefIE}} \text{Count}\left(\text{gram}_2\right)} \quad (2)$$

C. Task Evaluation

To verify the effectiveness of large language models in the traffic event extraction task, this paper compares several advanced open-source and closed-source large models. The basic information of the models is as follows:

Baichuan4 [27]: The new generation foundation model released by Baichuan Intelligence.

Qwen-max [21]: The ultra-large-scale language model Qwen-Max by Tongyi Qianwen.

GLM4 [28]: The foundation model GLM-4-0520 by Zhipu AI.

Qwen1.5-14B-Chat: A dialogue language model with 14 billion parameters.

OneKE [26]: A large-scale knowledge extraction model based on Chinese-Alpaca-2-13B, fully fine-tuned on the IEPile dataset.

Qwen2-7B-Chat: A dialogue language model with 7 billion parameters.

TRKE: The model proposed in this paper, based on Qwen1.5-14B-Chat fine-tuned on the IEPile and traffic event datasets.

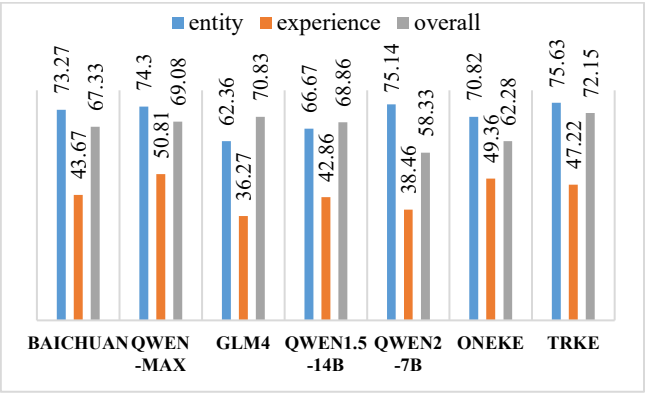
First, experiments were conducted on closed-source general-purpose large models, using the modular prompt strategy to guide these models in completing the traffic event extraction task. Since these large models have a large number of parameters and are pre-trained on rich corpora, they already possess excel-lent natural language processing capabilities. According to the experimental results shown in Table IV, these large models can initially complete the traffic event extraction task under the guidance of prompt strategies. Among them, the Qwen-Max model performed the best in extracting both travel element arguments and experience sentence arguments compared to other models. Therefore, the Qwen-Max model was selected for generating the fine-tuned dataset in subsequent experiments.

A total of 3,466 traffic guide texts from Liaoning Province were selected as input texts, and the few-shot learning capabilities of the large model were used to build a traffic

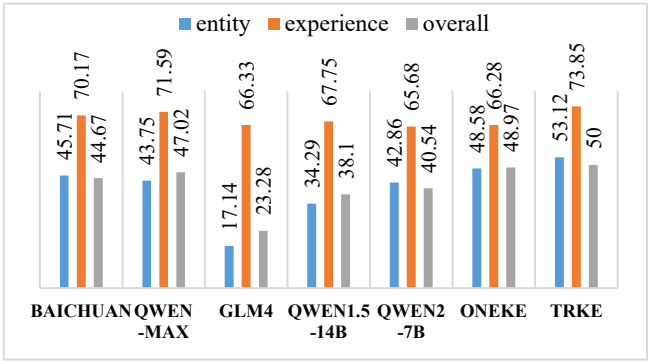
event dataset. After preliminary extraction by the Qwen-Max model and manual correction, the final fine-tuned dataset was generated, which contained a total of 7,138 traffic events, including 39,503 travel element arguments and 15,864 travel experience arguments. The distribution of key arguments is shown in Table 4.

Table 4 Arguments in the Fine-Tuning Dataset

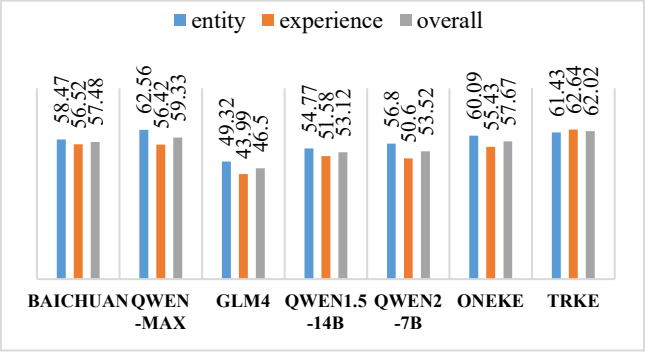
Argument Type	Argument Name	Quantity
Travel Elements	City	6558
	Departure Location	4303
	Destination	6145
	Mode of Transportation	7023
Travel Experience	Travel Experience	1269
	Notes and Suggestions	2503



(a) Accuracy



(b) Recall



(c) F1 score

Fig. 4 Traffic event extraction test results (unit: %)

For dataset partitioning, 80% of the dataset was used as the training set, 10% as the test set, and 10% as the validation set. To avoid errors in the experimental results, all models were tested 5 times, and the average of all results was taken as the final result. The specific test data and experimental results are shown in Figure 4. Compared with other models, the model proposed in this paper performed the best in the traffic event extraction task, with the TRKE model achieving an overall F_1 score of 62.02%. Compared with the Qwen1.5-14B-Chat and Qwen2-7B-Chat models that were not trained on the general information extraction dataset IEPile, the overall performance improved by 8.5%, proving that general information extraction training can effectively improve the model's traffic event extraction capability. Compared to the OneKE model, the TRKE model performed slightly better due to differences in pre-training data and training methods. Compared with the Qwen-Max model, the F_1 score for travel element argument extraction and traffic experience argument extraction improved by 2.25% and 3.33%, respectively, and the overall argument F_1 score improved by 2.88%. In summary, the experimental results show that large language models can accurately extract traffic route knowledge.

V. Conclusion

This paper designs a large language model-enhanced traffic route knowledge extraction model for traffic guide texts from social media. The large model is used to filter traffic guide texts, and a traffic event extraction template is developed based on the traffic event knowledge system. A fine-tuned traffic event dataset is constructed, and the Qwen model is instruction fine-tuned using this dataset, resulting in the traffic route extraction model TRKE. Experimental results demonstrate that, in the task of traffic event extraction, fine-tuning a small-parameter large model achieves higher accuracy compared to directly using a general-purpose large model for information extraction. The extracted structured traffic route knowledge helps travelers create route planning solutions that better meet their actual travel needs. In future work, this paper will focus on improving the scale and quality of the fine-tuned dataset, standardizing the traffic route knowledge graph, and further using the traffic route knowledge graph to assist travelers in making more informed decisions.

REFERENCES

[1] Jin, G., Wang, M., Zhang, J., Sha, H., & Huang, J., "STGNN-TTE: Travel time estimation via spatial–temporal graph neural network," Future Generation Computer Systems, vol. 126, pp. 70–81, 2022, doi: 10.1016/j.future.2021.07.012.

[2] Liu, H., Li, T., Hu, R., Fu, Y., Gu, J., & Xiong, H., "Joint representation learning for multi-modal transportation recommendation," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 1, pp. 1036-1043, 2019, doi: 10.1609/aaai.v33i01.33011036.

[3] Fu, L., Sun, D., & Rilett, L. R., "Heuristic shortest path algorithms for transportation applications: state of the art," Computers & Operations Research, vol. 33, no. 11, pp. 3324-3343, 2006, doi: 10.1016/j.cor.2005.03.027.

[4] Liu, H., Tong, Y., Han, J., Zhang, P., Lu, X., & Xiong, H., "Incorporating Multi-Source Urban Data for Personalized and Context-Aware Multi-Modal Transportation Recommendation," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 2, pp. 723-735, 2022, doi: 10.1109/TKDE.2020.2985954.

[5] Wang, J., Chang, Y., Li, Z., et al., "TechGPT-2.0: A large language model project to solve the task of knowledge graph construction," 2024, arXiv:2401.04507.

[6] Wang, Y., Li, G., & Tang, N., "Querying Shortest Paths on Time Dependent Road Networks," Proceedings of the VLDB Endowment, vol. 12, no. 11, pp. 1249-1261, 2019, doi: 10.14778/3342263.3342265.

[7]Gao, L., Chen, C., Chu, F., Liao, C., & Huang, H., "MOOP: An Efficient Utility-Rich Route Planning Framework Over Two-Fold Time-Dependent Road Networks," IEEE Transactions on Emerging Topics in Computational Intelligence, 2023, doi: 10.1109/TETCI.2023.3241930.

[8] Chen, Z., Shen, H. T., & Zhou, X., "Discovering Popular Routes from Trajectories," in Proceedings of the 27th IEEE International Conference on Data Engineering, pp. 900-911, 2011, doi: 10.1109/ICDE.2011.5767890.

[9] Tian, W., Shi, J., Luo, S., Li, H., Xie, X., & Zou, Y., "Effective and Efficient Route Planning Using Historical Trajectories on Road Networks," Proceedings of the VLDB Endowment, vol. 16, no. 10, pp. 2512-2524, 2023, doi: 10.14778/3603581.3603591.

[10] Cui, G., Luo, J., & Wang, X., "Personalized travel route recommendation using collaborative filtering based on GPS trajectories," International Journal of Digital Earth, vol. 11, no. 3, pp. 284–307, 2018, doi: 10.1080/17538947.2017.1326535.

[11] Wang, J., Wu, N., Zhao, W. X., Peng, F., & Lin, X., "Empowering A* Search Algorithms with Neural Networks for Personalized Route Recommendation," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'19), pp. 539–547, 2019, doi: 10.1145/3292500.3330824.

[12] Liu, H., Li, Y., Fu, Y., Mei, H., & Xiong, H., "Polestar++: An Intelligent Routing Engine for National-Wide Public Transportation," IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 6, pp. 6194-6208, 2023, doi: 10.1109/TKDE.2022.3153711.

[13] Roman, Y., "Scenario Customization for Information Extraction," PhD thesis, New York University, New York, USA, 2000.

[14] McCallum, A., & Li, W., "Early results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons," in Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, vol. 4, pp. 188–191, 2003, doi: 10.3115/1119176.1119206.

[15] Zhou, G. D., Su, J., Zhang, J., & Zhang, M., "Exploring Various Knowledge in Relation Extraction," in Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), pp. 427-434, Ann Arbor, Michigan, USA, June 2005, doi: 10.3115/1219840.1219893.

[16] Chieu, H. L., & Ng, H. T., "A Maximum Entropy Approach to Information Extraction from Semi-Structured and Free Text," in Proceedings of the 18th National Conference on Artificial Intelligence (AAAI-02), pp. 786-791, 2002.

[17] Yang, H., Chen, Y., Liu, K., Xiao, Y., & Zhao, J., "DCFEE: A Document-level Chinese Financial Event Extraction System based on Automatically Labeled Training Data," in Proceedings of ACL 2018, System Demonstrations, pp. 50-55, Melbourne, Australia, July 2018, doi: 10.18653/v1/P18-4009.

[18] Li, F., Peng, W., Chen, Y., Wang, Q., Pan, L., Lyu, Y., & Zhu, Y., "Event Extraction as Multi-turn Question Answering," in Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 829-838, November 2020, doi: 10.18653/v1/2020.findings-emnlp.73.

[19] Brown, T. B., Mann, B., Ryder, N., et al., "Language Models are Few-Shot Learners," 2020, arXiv:2005.14165.

[20] Touvron, H., Lavril, T., Izacard, G., et al., "LLaMA: Open and Efficient Foundation Language Models," 2023, arXiv:2302.13971.

[21] Bai, J., Bai, S., Chu, Y., et al., "Qwen Technical Report," 2023, arXiv:2309.16609.

[22] Xu, D., Chen, W., Peng, W., et al., "Large Language Models for Generative Information Extraction: A Survey," 2023, arXiv:2312.17617.

[23] Bi, Z., Chen, J., Jiang, Y., Xiong, F., Guo, W., Chen, H., & Zhang, N., "CodeKGC: Code Language Model for Generative Knowledge Graph Construction," ACM Transactions on Asian and Low-Resource Language Information Processing, 2024, doi: 10.1145/3641850.

[24] Taori, R., Gulrajani, I., Zhang, T., et al., "Alpaca: A Strong, Replicable Instruction-Following Model," 2023, Stanford Center for Research on Foundation Models.

[25] Hu, E. J., Shen, Y., Wallis, P., et al., "LoRA: Low-Rank Adaptation of Large Language Models," 2021, arXiv:2106.09685.

[26] Gui, H., Ye, H., Yuan, L., et al., "IEPile: Unearthing Large-Scale Schema-Based Information Extraction Corpus," 2024, arXiv:2402.14710.

[27] Yang, A., Xiao, B., Wang, B., et al., "Baichuan 2: Open Large-scale Language Models," 2023, arXiv:2309.10305.

[28] GLM Team, Zeng, A., Xu, B., Wang, B., et al., "ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools," 2024, arXiv:2406.12793.

You Hangyu (1999-), male, master. His research focuses on artificial intelligence.

Sun Huanliang(1969-), male, professor. His research focuses on artificial intelligence and data mining.

Route Planning Algorithm in Spatio-temporal Heat Map

Wang Yaxing Ma Xiaohui

Liaoning Province Big Data Management and Analysis Laboratory of Urban Construction, Shenyang 110168, Liaoning, China

***Abstract*—**The spatio-temporal heat map of the population reflects the distribution of the population in different time and space. People have the following needs for path queries: during the epidemic, people may want to avoid areas with high density of people, or those traveling on foot may be willing to avoid poor air quality road sections. This paper proposes a route planning problem in a crowd spatio-temporal heat map. Given the starting and ending points and travel length cost budget, it returns a route that has the minimum length through a certain heat map level (known as a dangerous road section) and the total cost does not exceed the travel length cost budget. We design a two-stage algorithm that first searches for the shortest path between the starting and ending points as the initial solution, and then replaces the included dangerous road sections. To reduce the time and space costs in the path search process, we propose an A * algorithm based on road hazard level filtering, which optimizes the search space and improves search efficiency. For a more effective measurement between the length of road sections and the safety and danger attributes, based on the A * algorithm, considering the quantitative relationship between travel length cost budget and shortest path cost, the evaluation function is improved, and a weight method for determining the evaluation function is proposed to fully optimize the length of dangerous road sections in the path. Finally, experiments were conducted on real datasets to verify the validity of the proposed algorithm.

Keywords—Route Planning; Spatio-temporal Heat Map; Road Network Constraints; Dangerous Road Section; Travel Length Cost Budget

I. INTRODUCTION

The increasing popularity of electronic devices equipped with location-based functions, such as in-car navigation systems and smartphones, a vast amount of location information has been generated from their widespread use, making it possible to fulfill path queries catering to various preferences. For instance, there are optimal path queries with keyword awareness [1], finding the least costly routes within safe and preferred areas [2], and selecting the least risky paths under time-varying and uncertain travel costs [3].

A heatmap, visualized through a density function, represents the density of points on a map, enabling people to perceive the density of points independently of zoom factors. Heatmaps have been widely applied in multiple fields: a heatmap of urban population distribution can intuitively reflect the density of crowds in various locations within a city, while an air quality heatmap can indicate the air quality status of a region.

During necessary travel during epidemic prevention and control, choosing a less crowded route to reduce contact between individuals has become a practical need. Hikers are more inclined to avoid areas with polluted air and seek travel routes with better air quality. Based on these application requirements, this paper proposes the problem of path planning in spatiotemporal heatmaps. Given a heatmap, the starting and ending points of a trip, and a travel length cost budget for the path, path planning in a spatiotemporal heatmap involves finding a path from the starting point to the ending point such that the total length of the path does not exceed the travel length cost budget and the length of hazardous road segments with a heat level greater than or equal to level k (where $1 < k \leqslant \theta_{max}$, and θ_{max} is the highest heat level) is minimized.

Figure 1 illustrates an example of path planning in a spatiotemporal heatmap, where s represents the starting point and d represents the ending point. Three paths are shown in the figure: Path S1 is the shortest path (i.e., Euclidean distance) between the starting and ending points; Path S2 is the path with the lowest crowd density between the starting

and ending points. It is clearly visible from the figure that most of the areas this path traverses are purple, blue, and green, indicating relatively low heat levels; Path S3 is the path with the highest crowd density between the starting and ending points, passing through areas with the highest heat levels.

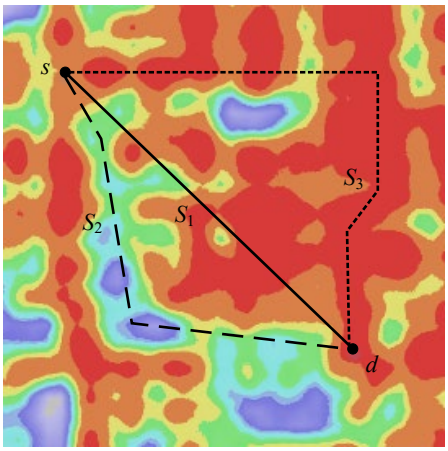


Fig. 1 Route planning in spatio-temporal heat map

Due to the complexity and large volume of data involved in the path planning problem studied in this paper, which encompasses road networks, heatmaps, and the locations of crowds, path planning in spatiotemporal heatmaps is challenging. This challenge mainly lies in improving the accuracy and efficiency of queries during the path search process under the constraints of spatiotemporal heatmaps.

To address these challenges, this paper designs a two-phase query algorithm. Initially, the shortest path between the starting and ending points is searched as the initial solution, and then the hazardous road segments within it are replaced with safe ones. To reduce the time and space costs of traversing nodes during the path search process, this paper proposes an A* algorithm based on road segment hazard level filtering. Building upon the traditional A* algorithm, before a node enters the OPEN list, it is screened and optimized based on the heat levels and lengths of its adjacent edges, thus avoiding unnecessary space searches and improving search efficiency. To balance the length and safety attributes of road segments more effectively, an improved evaluation function is introduced in the A* algorithm, considering the quantitative relationship between the travel length cost budget and the shortest path cost. Additionally, multiple methods are proposed to determine the weight w in the evaluation function, minimizing the length of hazardous road segments in the path.

The main contributions of this paper are as follows:

(1) Defining the path planning problem in spatiotemporal heatmaps;

(2) Designing an A* algorithm based on hazardous road segment replacement, proposing an A* algorithm based on road segment hazard level filtering, and an A* algorithm utilizing road segment hazard levels to adjust the evaluation function;

(3) Conducting experimental comparisons on real datasets to assess the quality of query results and the efficiency of the algorithms, verifying the effectiveness of the proposed algorithms.

II. Introduction

Related work includes shortest path queries and preference path queries.

A. Shortest path query

The shortest path query is the most basic problem in path planning, and it is usually optimized by pre-computing and storing various auxiliary information for query optimization. The literature [4] focuses on the Hub Labeling (HL) technology, which shortens the indexing time in shortest path queries. A new labeling scheme based on graph independent sets was proposed in reference [5], which can handle path queries in massive graph data. The literature [6] noticed the application of caching technology in online shortest path queries and designed a compact caching structure that supports efficient query matching operations. In reference [7], a hierarchical 2-hop index was proposed, along with an algorithm for constructing an h2h index based on a distance-preserving graph, resulting in optimized index size and execution time. The literature [8] proposes the Petal A-1N algorithm, which processes 1-N shortest path queries in batches and uses a zigzag scheduling method to answer the entire query set, further reducing the total running time. Inspired by R-Tree, literature [9] proposed a highly balanced and scalable index G-Tree that can effectively support shortest path queries.

The literature [10] studies the maintenance of shortest path indexes in dynamic road networks and proposes a method of contraction hierarchy as a shortest path calculation method, which has excellent preprocessing time, space cost, and overall query performance. The literature [11] studies the Group Nearest Neighbor (GNN) query and uses R-tree indexing to solve this problem. The literature [12] focuses on group travel plan queries, where the group has an interest in minimizing the total travel distance of all members. The literature uses an iterative method based on the group nearest neighbor query idea and an effective hierarchical method to evaluate the query with a single pass through the R tree.

The path length obtained in this query is not the shortest,

and the total cost of the path should not exceed the given travel length cost budget, while also minimizing the length of dangerous road sections.

B. Preference path query

The purpose of preference path query is to meet the personalized needs of users for path query. Typical preference path queries include arc orientation problems [13-16], keyword coverage path queries [1,17-19], and time-dependent path queries [3,20-22].

The goal of the Arc Orienteering Problem (AOP) is to query a path that maximizes the total score of the target within a given cost budget. It is a typical NP-hard combinatorial optimization problem. The literature [13] proposes and evaluates the Greedy Random Adaptive Search Algorithm, which is a meta-heuristic method that can solve AOP instances to near-optimal states within seconds of execution time. The literature [14] studies the variant of the AOP problem of ride planning and proposes a branch-cut method and an iterative local search algorithm to solve the problem. In reference [15], the road network is regarded as a spatial network, and a series of meta-heuristic methods are proposed to support fast response time for large-scale road networks using elliptic pruning and spatial indexing techniques in the field of spatial databases. The literature [16] studies the problem of finding the route with the highest cumulative scenery score within a given travel time budget, and proposes a memetic algorithm to achieve path planning.

Keyword coverage path query retrieves the paths covered by the query keywords specified by the user. The literature [1] proposes a keyword-aware optimal path query, which seeks to find an optimal path that covers a set of user-specified keywords, satisfies specified budget constraints, and maximizes the target score. The literature [17] noticed that users have different preference weights for specific keywords in path queries, and proposed an optimization for keyword coverage to achieve the optimal coverage range of keywords. In reference [18], a method is proposed for conducting multi-approximate keyword path searches within GIS data, utilizing multiple keywords and approximate string similarity functions to supplement the shortest path search. The literature [19] focuses on querying routes of interest on a road network. This query allows users to specify their interests by querying keywords and returns a path with a distance threshold less than the maximum distance and the largest number of keywords.

Time-dependent path queries are path queries that consider time costs. The literature [3] focuses on the fact that the travel time cost of a path will change over time and is uncertain. Therefore, the time series method is used to divide time into multiple intervals, and within each interval, the path is queried using the non-dominated lowest cost. The literature [20] focuses on the problem of probabilistic budget path, which is to find the path with the highest probability of reaching the destination within a given time budget. A time-dependent acceleration technique for uncertain contraction hierarchy is proposed in reference [21] to address the rapid path planning problem with time-varying and uncertain edge weights. A dynamic discrete discovery algorithm is proposed in reference [22] to minimize the time required for travel on the arc and to minimize the total time from the starting point to the end point.

The research in this paper is a typical preference path query, which divides safe and dangerous road sections based on the thermal level of edges in the road network, and determines the preference level for safe and dangerous road sections based on the size of the travel length cost budget. The path query proposed in this paper is a variant of the AOP problem. Each edge in the road network is associated with a cost, which is the length of the edge, and a score, which is the thermal level of the edge. The objective of this study is to find a route with the minimum length of dangerous road sections, within the constraints of a given travel length cost budget.

III. Problem Definitions

Definition 1: Thermal Level The thermal level, denoted as θ, represents the intensity of a certain characteristic within a given range. It is expressed using 1, 2, …, θ_{max}, where 1 signifies the lowest level and θmax signifies the highest level. In the context of a thermal map of urban population distribution, the population density is categorized into 7 distinct thermal levels, represented by gray, purple, blue, green, yellow, orange, and red in ascending order.

Definition 2: Road Network with Thermal Levels Given a graph G(V, E), where V is the set of nodes and E is the set of edges. The total number of nodes is denoted by |V|, and each node v∈V represents an intersection of edges. The total number of edges is denoted by |E|, and each edge is represented as ei (<vi, vj>, θ), where vi and vj are the starting and ending points of the edge respectively, and θ is the thermal level of that edge. The length of the edge is denoted as ei.l.

Definition 3: Spatial Route Planning in Road Networks under Temporal-Spatial Thermal Map Constraints Given a road network G<V, E> with thermal levels, a starting point s, an endpoint d, a travel length cost budget B, the spatial route

planning Q(s, d, B, k) in road networks under temporal-spatial thermal map constraints involves finding a path S from the starting point s to the endpoint d on the road network with thermal levels, such that the total cost S.l of the path S does not exceed the budget B, and the length L of road segments with a thermal level of at least k is minimized. L is calculated using Equation 1.

$$L = \sum_{e_i \in S,\ k \le e_i.\theta \le n} e_i.l \quad (1)$$

The length L of hazardous road segments with a thermal level of at least k in path S is less than the L values of all other paths Sn from the starting point s to the endpoint d that satisfy the travel length cost budget B, as shown in Equation 2.

$$L(S) < L(S_n) \quad (2)$$

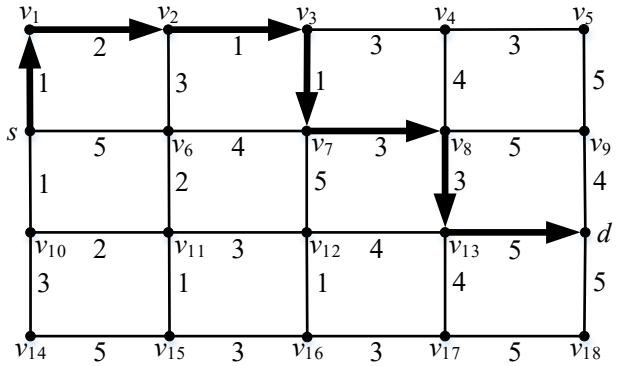


Fig. 2 Example of spatial route planning of road network under constraints of spatio-temporal heat map

Figure 2 demonstrates an example of the spatial route planning algorithm in road networks under temporal-spatial thermal map constraints. Assuming the road network G has thermal levels ranging from 1 to 5, with the numbers marked on the edges representing the thermal levels. Suppose the cost of all edges in the figure is 1. The query is to find a path from s to d that minimizes the length of hazardous road segments with a thermal level of at least 4, while satisfying the cost budget constraint of B = 7. The final query result is the path s→ v1→ v2→ v3→ v7→ v8→ v13→ d, which has a total cost of 7 and a hazardous road segment length of 1 with a thermal level of at least 4.

The problem of spatial route planning in road networks under temporal-spatial thermal map constraints studied in this paper is a variant of the AOP (Arc Routing Problem) problem. In this problem, each edge is associated with not only a cost but also a thermal level. Assuming that each node has an out-degree of λ, meaning there are λ routes to choose from after passing through the node, and the number of nodes that can be traversed within the travel length cost budget is t, then the cost is λt. Therefore, the problem of spatial route planning in road networks under thermal map constraints is an NP-hard problem.

Table 1 briefly summarizes the symbols used in this paper.

TABLE 1 SYMBOL SUMMARY

Symbols	Description
$G(V, E)$	Road Network with Thermal Levels
v_i	A Node on Road Network G
$e_i(<v_i, v_j>, \theta)$	An Edge on Road Network G
θ	Thermal Level of an Edge
$e_i.l$	Length of an Edge
$Q(s, d, B, k)$	Spatial Route Planning in Road Networks under Temporal-Spatial Thermal Map Constraints

IV. QUERY PROCESSING

This section introduces two proposed query processing algorithms, including an A* algorithm based on road segment danger level filtering and an A* algorithm that adjusts its evaluation function using road segment danger levels.

A. A Algorithm Based on Road Segment Danger Level Filtering*

The A* algorithm is a typical heuristic algorithm that estimates the minimum cost from a node v to the destination through an evaluation function f(v) = g(v) + h(v), guiding the algorithm to search consistently towards the destination to find the shortest path. In the A* algorithm, two lists need to be created: the OPEN list to store all generated but unexamined nodes, and the CLOSE list to store visited nodes. During the pathfinding process of the A* algorithm, an important step is to find all adjacent nodes of the current node and put them into the OPEN list. Afterward, the evaluation function f(v) is used to calculate the minimum cost estimate for each node in the OPEN list, and the nodes are sorted based on their estimates. The node with the smallest estimate is found, put into the CLOSE list, and considered for further processing.

In the traditional A* algorithm for finding the shortest path, distance is the only heuristic factor. However, in the problem addressed in this paper, not only must the total cost of the path not exceed the travel length cost budget, but also the length of dangerous road segments should be minimized. Therefore, danger levels are added as another heuristic factor. This paper proposes an A* algorithm based on road segment danger level filtering to reduce the number of candidate nodes in the OPEN list, optimize the time and space cost of sorting nodes in the OPEN list based on their evaluation values, and decrease the length of dangerous road segments in the path. Additionally, based on the traditional A* algorithm, an OPEN-OTHER list is introduced to store nodes that have been screened out and not entered the OPEN list. When

there is a risk of a dead-end due to nodes not entering the OPEN list, the OPEN-OTHER list is activated to allow the path search process to continue.

The specific process is shown in Algorithm 1. First, three lists, OPEN, OPEN-OTHER, and CLOSE, are established. The OPEN list is used to store all generated but unexamined nodes, the OPEN-OTHER list stores all nodes that have been screened out and not entered the OPEN list, and the CLOSE list stores visited nodes (Line 1). Then, the evaluation function value of the starting point s is calculated, and it is put into the OPEN list (Lines 2-3). Next, a loop is performed until the destination d enters the CLOSE list (Line 4). The node with the smallest evaluation function value in the OPEN list at this time is set as the current node v and moved from the OPEN list to the CLOSE list (Lines 5-6). At this point, the adjacent nodes of the parent node of the current node v are searched in the OPEN-OTHER list, and the adjacent node with the shortest danger length is set as the new current node v, and the pathfinding process continues at Line 11. If there is no risk of a dead-end, the adjacent nodes of the current node v are traversed. If the danger level of the edge connecting an adjacent node and the current node v is less than level k (i.e., the edge is safe), the adjacent node is put into the OPEN list (Lines 10-13). If the danger level of the edge is greater than or equal to level k (i.e., the edge is dangerous), all dangerous edges associated with the current node are compared in terms of length, and the edge with the smallest length is selected. The node connected by this edge is put into the OPEN list. Nodes connected by edges with longer danger lengths are put into the OPEN-OTHER list for backup and will not enter the OPEN list for comparison based on evaluation function values (Lines 14-17). After completing the judgment of whether all adjacent nodes of the current node v enter the OPEN list, the evaluation function values of each node in the OPEN list at this time are calculated and sorted in ascending order to prepare for the next loop (Line 18). When the destination d enters the CLOSE list, the loop terminates. Starting from the destination d in the CLOSE list, the parent nodes are found sequentially until the starting point s is reached, resulting in path S (Line 19). Finally, it is judged whether the total cost of S satisfies the constraint of the travel length cost budget. If the budget is not exceeded, path S is returned as the final solution. Otherwise, the path search does not have a qualified result for this time (Lines 20-21).

Algorithm 1 A* Algorithm with Road Segment Danger Level Filtering

Input: $Q(s, d, B, k)$, G

Output: Path S with total cost ≤ B and minimized length of dangerous road segments with θ ≥ k

(1) Create OPEN, OPEN-OTHER, CLOSE lists;
(2) Calculate the heuristic value of the start node s;
(3) Add s to the OPEN list;
(4) **while** (d is not in CLOSE)
(5) Select the node v with the smallest heuristic value from the OPEN list;
(6) Move v from the OPEN list to the CLOSE list;
(7) **if** (v has no adjacent nodes except its parent)
(8) Search for all adjacent nodes of v's parent in OPEN-OTHER, and set the one with the shortest dangerous length as v;
(9) **Go to** (11);
(10) **else**
(11) **for** (each adjacent node of v)
(12) **if** (the θ value of the edge connecting v to this adjacent node is < k)
(13) Add this adjacent node to the OPEN list;
(14) **else if** (this edge is the shortest among all edges connecting v to its adjacent nodes with θ ≥ k)
(15) Add this adjacent node to the OPEN list;
(16) **else**
(17) Add this adjacent node to the OPEN-OTHER list;
(18) Sort all nodes in the OPEN list by their heuristic values in ascending order;
(19) In the CLOSE list, starting from d, trace back to find parent nodes until reaching s, forming path S;
(20) **if** (the length of S, S.l, is ≤ B)
(21) **return** S;

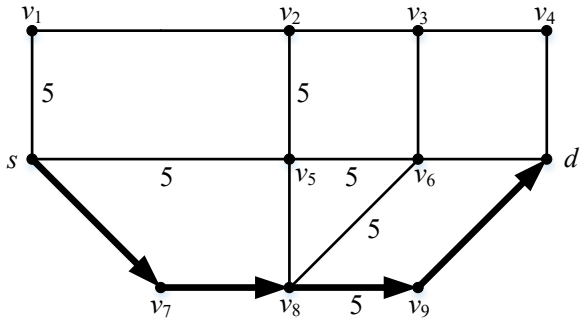


Fig. 3 The example of filtering algorithm based on road segment danger level

Figure 3 presents an example of an algorithmic query. Here, 's' represents the start point, and 'd' represents the destination. The costs of edges <s, v7>, <v8, v6>, and <v9, d> are √2, while the costs of edges <s, v5> and <v1, v2> are 2. The costs of all other edges are 1. The travel length cost budget B is set to 5. Assuming there are five heat levels ranging from 1 to 5, with level 5 being

critical, the start point 's' is first placed in the CLOSE list and set as the parent node. 's' has three adjacent nodes: v1, v5, and v7. Among the three edges connecting them to 's', <s, v1> and <s, v5> have a heat level of 5, making them critical edges. Since <s, v1> is shorter, it is retained, and <s, v5> is discarded. Therefore, adjacent nodes v1 and v7 are added to the OPEN list, while v5 is discarded. Due to v7 having a smaller evaluation function value, it is transferred from the OPEN list to the CLOSE list, set as the new parent node, and all its unvisited adjacent nodes, v8, are added to the OPEN list. At this point, v8 in the OPEN list has the smallest evaluation function value, so it is transferred to the CLOSE list and set as the new parent node. v8 has three adjacent nodes: v5, v6, and v9. Among them, <v8, v6> and <v8, v9> are critical edges with a heat level of 5. Since <v8, v9> is shorter, it is retained, and <v8, v6> is discarded. Adjacent nodes v5 and v9 are added to the OPEN list, while v6 is discarded. At this point, v9 in the OPEN list has the smallest evaluation function value, so it is transferred to the CLOSE list and set as the new parent node. Since v8 has only one adjacent node, 'd', the path search reaches the destination, and 'd' is added to the CLOSE list. Starting from 'd', parent nodes are sequentially traced back in the CLOSE list until the start point 's' is found, resulting in the final path s→v7→v8→v9→d with a total cost of $2+2\sqrt{2}$.

The algorithm proposed in this section calculates and compares the length and heat level of the edge connecting a node to its parent node before placing it in the OPEN list, filtering out longer critical edges. When the number of critical edges in the road network is relatively low, this algorithm performs well. Since the proposed algorithm screens and filters nodes before placing them in the OPEN list, the number of nodes entering the OPEN list for evaluation function value comparison is less than in the traditional A* algorithm.

B. A Algorithm with Evaluation Function Adjusted Based on Road Segment Danger Levels*

During the path search process, it is necessary to avoid road segments with a danger level of k or higher as much as possible. Therefore, the core of solving the problem presented in this paper lies in choosing between dangerous and safe road segments: a shorter safe road segment is always the preferred option, but it needs to be assessed which is more beneficial for obtaining the desired result when comparing a longer safe road segment with a shorter dangerous one.

The traditional A* algorithm can quickly and effectively solve the shortest path problem in static road networks. However, when used to solve the problem presented in this paper, it lacks a comparison between safe and dangerous road segments. Therefore, this section proposes an A* algorithm that adjusts the evaluation function based on road segment danger levels, combining the greedy strategy to solve the selection problem between safe and dangerous road segments. The evaluation function for a node is shown in Equation (3).

$$f(v) = g(v) + h(v) + w*lv \quad (3)$$

In this equation, g(v) and h(v) are the same as in the traditional A* algorithm. g(v) represents the shortest path length on the road network from the start point to node v, serving as the actual cost from the start point to node v. h(v) represents the Euclidean distance from node v to the destination, serving as the estimated cost from node v to the destination. w*lv is the estimated cost proposed in this section, where lv is the length of the edge connecting node v to its parent node, and w is a weight. The value of w depends on two cases: if the danger level of the edge connecting node v to its parent node is less than k (i.e., a safe edge), then w = 0; if the danger level of the edge is k or higher (i.e., a dangerous edge), then w is a positive integer. The determination of the w value will be discussed later in this section.

The specific process is shown in Algorithm 2. First, an OPEN list is established to save all generated but unexamined nodes, and a CLOSE list is established to save visited nodes (Line 1). Then, the start point s is placed in the OPEN list (Line 2). Next, the pathfinding stage begins until the destination d is reached (Line 3). The node with the smallest evaluation function f value in the OPEN list at this time is set as the current node v and transferred from the OPEN list to the CLOSE list (Lines 4-5). The adjacent nodes of the current node v are traversed and placed in the OPEN list (Lines 6-7). If the danger level of the edge connecting an adjacent node to the current node v is less than k, the evaluation function for that adjacent node is f(v) = g(v) + h(v) + w * lv, with w = 0 (Lines 8-9); if the danger level of the edge is k or higher, the evaluation function for that adjacent node is f(v) = g(v) + h(v) + w * lv, with w > 0 (Lines 10-11). After traversing all adjacent nodes of the current node v, each node in the OPEN list is sorted in ascending order based on the f value, and the next loop begins (Line 12). The loop terminates when the destination d is placed in the CLOSE list. In the CLOSE list, starting from the destination d, the parent nodes are sequentially traced back until the start point s is found, resulting in path S (Line 13). Finally, it is judged whether the total cost of S satisfies the constraint of the travel length cost budget. If it does not exceed the budget, path S is returned as the final solution; otherwise, the path search has no eligible result for this time (Lines 14-15).

Algorithm 2 A* Algorithm with Adjusted Evaluation Function Based on Road Segment Danger Level

Input: $Q(s, d, B, k)$, G

Output: Path S with total cost ≤ B and minimized length of dangerous road segments where θ ≥ k

(1) Create OPEN and CLOSE lists;
(2) Add s to the OPEN list;
(3) **while** (d is not in CLOSE)
(4) Retrieve node v with the smallest evaluation value f from the OPEN list;
(5) Move v from the OPEN list to the CLOSE list;
(6) **for** (each adjacent node of v)
(7) Add the adjacent node to the OPEN list;
(8) **if** (θ of the edge connecting v and the adjacent node < k)
(9) Calculate f(v) = g(v) + h(v) + w * lv, where w = 0;
(10) **else**
(11) Calculate f(v) = g(v) + h(v) + w * lv, where w > 0;
(12) Sort all nodes in the OPEN list in ascending order based on their f values;
(13) In the CLOSE list, trace back from d to find the parent nodes to obtain path S;
(14) **if** (S.l ⩽ B)
(15) **return** S;

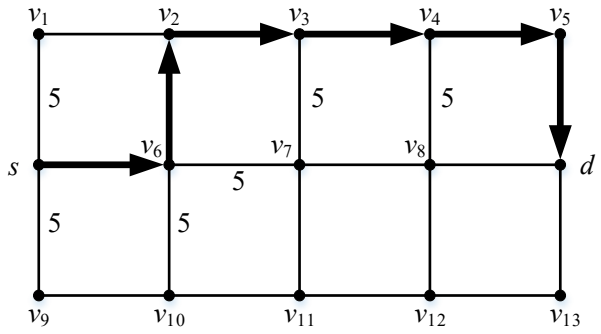


Fig. 4 The example of adjustment evaluation function algorithm

Figure 4 illustrates an example of the algorithm that adjusts the evaluation function. 's' represents the starting point, and 'd' represents the destination. Assuming there are five heat levels ranging from 1 to 5, with level 5 being the most dangerous. The weight 'w' is set to 10. Initially, the starting point 's' is placed in the CLOSE list and designated as the parent node. All unvisited adjacent nodes of 's', namely v1, v6, and v9, are added to the OPEN list, and their evaluation function values are calculated. Since edges <s, v1> and <s, v9> are dangerous, when 'w' is 10, the evaluation function values of v1 and v9 will be greater than that of v6. Therefore, v6 is transferred from the OPEN list to the CLOSE list, set as a new parent node, and its unvisited adjacent nodes v2, v7, and v10 are added to the OPEN list. Edges <v6, v7> and <v6, v10> are also dangerous, so the evaluation function values of v6 and v10 will be greater than that of v2. After comparison, v2's evaluation function value is smaller than those of v1 and v9, which were added to the OPEN list in the previous step. Thus, v2 is currently the node with the smallest evaluation function value in the OPEN list. It is then transferred to the CLOSE list, and all its unvisited adjacent nodes are added to the OPEN list with their evaluation function values calculated and compared. This process continues until the destination 'd' is placed in the CLOSE list, ending the pathfinding stage. Starting from 'd', parent nodes are traced back in the CLOSE list until the starting point 's' is reached, resulting in the final path marked by the thick black arrowed line in the figure: s→ v6→ v2→ v3→ v4→ v5→ d. The total cost of this path is 6, and it contains no dangerous segments.

The algorithm proposed in this section increases the estimated cost related to the length of dangerous edges. Adjusting the value of 'w' can alter the pathfinding direction, effectively avoiding dangerous road segments. The average time complexity of this algorithm is the same as that of the A* algorithm, denoted as O(NlogN). In this algorithm, two heuristic factors are considered: distance and heat level. The distance heuristic aims to find a shorter path to meet the constraint of the travel length cost budget, while the heat level heuristic aims to avoid dangerous road segments as much as possible, reducing the length of dangerous segments in the path.

Therefore, the value of the evaluation function f(v) is influenced by two estimated costs: h(v) and w*lv. To prioritize reducing the length of dangerous road segments in the path, the chosen path is not always the shortest but the additional length cost is limited, which is the reason for setting the travel length cost budget. Considering different scenarios, this paper proposes the following three methods to determine the value of 'w':

Precomputed Estimation Method: The core idea of this method is to find the relationship between the proportion of dangerous road segment length to the total path length and the weight 'w' to estimate its value. Given a starting point and a destination, the shortest path on the road network is searched, and the proportion 'p' of dangerous road segment length to the total path length is calculated. This proportion roughly reflects the distribution of dangerous road segments between the starting and destination points. Based on this, a weight 'w' is estimated to search for a new path that minimizes the length of dangerous road segments. This paper expects the relationship between proportion 'p' and weight 'w' to be as shown in Equation 4, where 'a' and 'b' are different positive integers.

$$w(p) = \begin{cases} a, & 0 < p \le 10\% \\ b, & 10\% < p \le 20\% \end{cases} \quad (4)$$

Static Estimation Method: When 'h(v)' is determined as

the Euclidean distance from node 'v' to the destination, the selection of weight 'w' becomes crucial. A larger 'w' indicates a stronger preference for safe road segments and a lower tolerance for dangerous ones; a smaller 'w' indicates a higher tolerance for dangerous road segments. Therefore, when the total cost of the path exceeds the travel length cost budget, 'w' should be reduced. Conversely, when there is some surplus in the budget, 'w' can be increased appropriately to reduce the length of dangerous road segments in the path.

The core idea of the static estimation method is to find the relationship between weight 'w' and the total path cost. Given a starting point and a destination, using the A* algorithm requires some additional length cost 'le' beyond the shortest path. This is the additional length when searching with 'w > 0' compared to the shortest path found using the A* algorithm between the same starting and destination points. The additional length cost 'le' has the following relationship with the Euclidean distance between the starting and destination points: the closer the Euclidean distance, the less additional length cost is usually paid; the farther the Euclidean distance, the more additional length cost is usually paid. Therefore, this paper expects a relationship between the additional length cost 'le', the Euclidean distance 'D(s, d)' between the starting and destination points, and the weight 'w' as shown in Equation 5, where 'a' and 'b' are different positive integers, 'dis' is a positive integer, and 'x' and 'y' are different positive integers.

$$w(D, l_e) = \begin{cases} a, & D(s, d) \le dis,\ 0 < l_e \le x \\ b, & D(s, d) > dis,\ 0 < l_e \le y \end{cases} \quad (5)$$

Dynamic Adjustment Method: The core idea of this method is to use the larger of the weights estimated by the static estimation method and the precomputed estimation method as the initial weight. Based on the relationship between the total path cost and the travel length cost budget, the weight is adjusted, and the total path cost after each weight adjustment is compared to the travel length cost budget. The most suitable weight is found such that the total path cost is as close as possible to but does not exceed the travel length cost budget. Since a larger weight usually results in a shorter length of dangerous road segments in the path, the weight obtained by the dynamic adjustment method is the closest to the ideal value among the three estimation methods. Set the initial weight to 'w0'. The method for adjusting the weight 'w' is as follows: First, if the total path cost does not exceed the travel length cost budget when the weight is adjusted to 'wu' (where 'u' is a positive integer), the next weight 'wu+1' is twice the current weight 'wu'. Second, if the total path cost exceeds the travel length cost budget for the first time when the weight is adjusted to 'wu', use the previous weight 'wu-1' as the left endpoint and the current weight 'wu' as the right endpoint of a closed interval [wu-1, wu]. Binary search is used within this interval to adjust the weight until the path with a weight of 'wu+c' (where 'c' is a positive integer) has a total cost that does not exceed the travel length cost budget, while the path with a weight of 'wu+c+1' exceeds it. The weight 'wu+c' is the final weight obtained using the dynamic adjustment method.

V. Experimental Analysis

This section mainly focuses on the experimental analysis of the proposed algorithms.

A. Dataset

The experimental dataset includes 63,923 road network data from Beijing, location data of people in Beijing for April and May 2017, and heatmap level data for Beijing's road network. The heatmap level of the road network is calculated based on people's location data and Baidu's urban heatmap. The heatmap of population distribution is divided into seven levels.

A. Comparison of Route Planning Algorithm Performance

For spatial route planning in the road network constrained by the spatio-temporal heatmap, two algorithms are implemented in this paper. Meanwhile, inspired by the ideas in literature [15], a two-stage path replacement algorithm is designed as a comparative algorithm.

AStarReplace: An A* algorithm based on dangerous road segment replacement, drawing on the replacement concept from the local iterative search algorithm in literature [15]. It is named AStarReplace as a comparative algorithm.

BaseOpen: An algorithm for spatial route planning in the road network constrained by the spatio-temporal heatmap. This algorithm optimizes the OPEN list of the traditional A* algorithm by screening the child nodes of the current node v before they enter the OPEN list, allowing safe road segments to pass the screening and proceed to the step of evaluating the cost function of subsequent nodes. For dangerous road segments, the longer ones are removed, leaving only the shortest route.

Greedy-Precom: An A* algorithm that adjusts the evaluation function using road segment danger levels and employs a pre-computation estimation method to determine the weight w.

Greedy-Static: An A* algorithm that adjusts the evaluation function using road segment danger levels and employs a static estimation method to determine the weight w.

Greedy-Dynamic: An A* algorithm that adjusts the evaluation function using road segment danger levels and employs a dynamic adjustment method to determine the weight w.

(1) Comparison of Dangerous Road Segment Lengths

This section compares the lengths of dangerous road segments for each algorithm under varying Euclidean distances between the origin and destination. The default setting for the danger level of road segments is level 7. When comparing different algorithms, the travel length cost budget is set in two scenarios: when the Euclidean distance between the origin and destination is no more than 7.5 km, the travel length cost budget is set as the shortest path length plus 0.5 km; when the Euclidean distance exceeds 7.5 km, the budget is set as the shortest path length plus 1 km. When comparing the weight changes of the Greedy algorithm alone, no travel length cost budget is set.

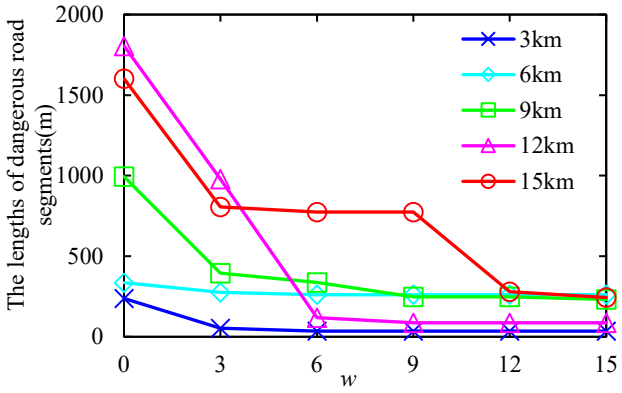


Fig. 5 Effects of weight *w*

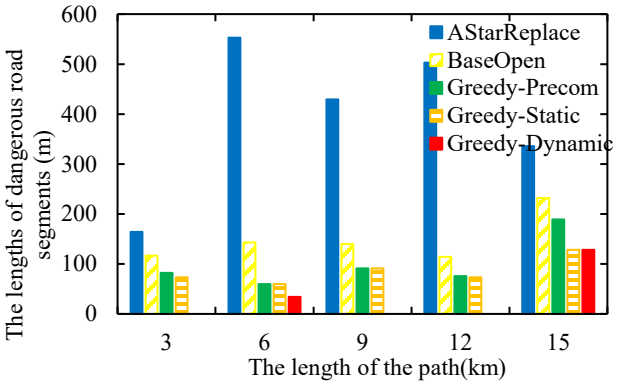


Fig. 6 Effects of the Euclidean distance between start and end points

Figure 5 compares the lengths of dangerous road segments for the Greedy algorithm under varying weights w. The Euclidean distances between the origin and destination are selected as five lengths ranging from 3 km to 15 km. As w increases, the lengths of dangerous road segments for each origin-destination pair show a decreasing trend. When w is a positive integer, it indicates a preference for safe road segments over dangerous ones. The larger the weight, the higher the preference for safe road segments. Although the length of dangerous road segments decreases overall as w increases, the lengths of dangerous road segments in the searched paths show a significant decrease. Figure 5 verifies the superiority of the Greedy algorithm in minimizing the length of dangerous road segments.

Figure 6 compares the lengths of dangerous road segments in the paths searched by the basic algorithm AStarReplace, the BaseOpen algorithm, and the Greedy algorithm using three different strategies to determine the weight w. The Euclidean distances between the origin and destination are selected as five lengths ranging from 3 km to 15 km. Compared to the basic algorithm AStarReplace, the lengths of dangerous road segments for the other four algorithms are significantly reduced. The Greedy algorithm outperforms the BaseOpen algorithm, indicating that the estimation and adjustment of the weight w are effective in avoiding dangerous road segments during the route-finding process. Among the three strategies for determining the weight w, the dynamic adjustment method performs best overall.

(2) Comparison of Algorithm Efficiency

This section compares the efficiency of various algorithms, with the default settings for the thermal ranking of hazardous road segments and the travel length cost budget assumed to be the same as those used in the comparison of hazardous road segment lengths in the previous section.

Figure 7 compares the time required for the Greedy algorithm to perform 10,000 path queries under varying weight values (w). It can be observed that for the same set of origin and destination points, the algorithm's running time increases as w increases. When w increases, the length of hazardous road segments in the searched path decreases, but at the cost of a certain increase in the total path cost. This increase in total cost translates to an increase in the number of nodes that need to be calculated and compared, resulting in an increase in the algorithm's running time.

Figure 8 compares the impact of the Euclidean distance between the origin and destination on algorithm efficiency, using the time required for 10,000 path queries as the metric. The BaseOpen algorithm reduces the number of nodes that enter the OPEN list for comparing evaluation function values, thereby improving operational efficiency. For the Greedy algorithm, regardless of the strategy used to determine weights, its search space is superior to that of the basic algorithm. Among the three weight optimization strategies for the Greedy algorithm, Greedy-Dynamic assigns larger weights than the other two strategies, resulting in a larger search space and thus requiring more running time.

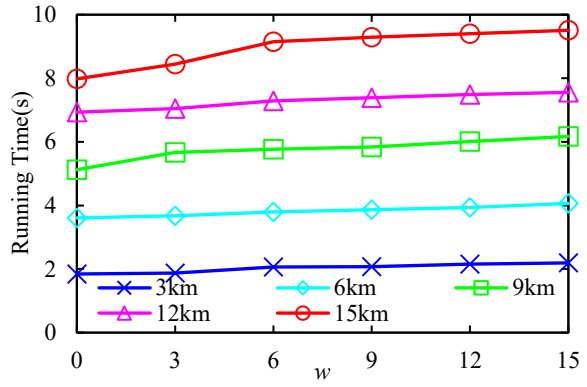


Fig. 7 Effects of weight *w* on the Greedy algorithm

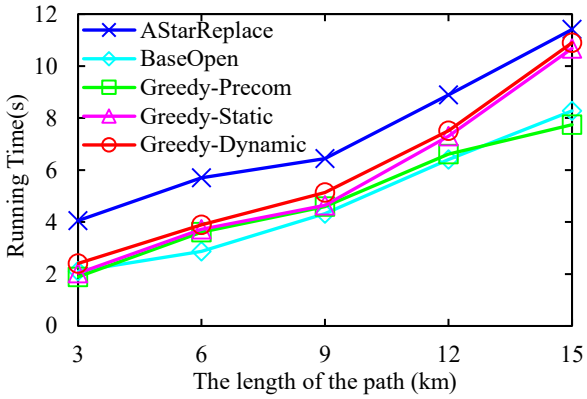


Fig. 8 Effects of the Euclidean distance between start and end points

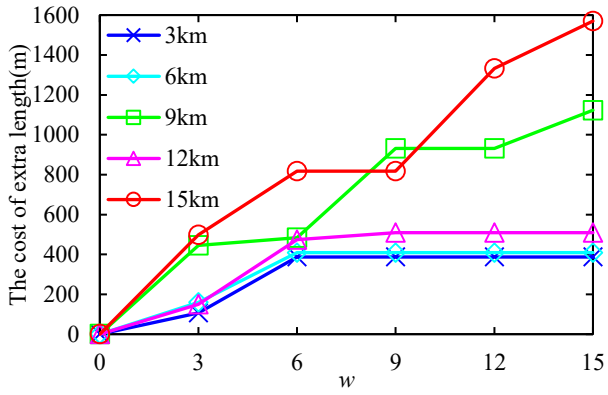


Fig. 9 Effects of weight *w* on the extra length cost

Figure 9 illustrates the impact of weights on the additional length cost in the Greedy algorithm. As the weights increase, path planning becomes increasingly biased towards safer road segments, which means that more additional length is required to avoid hazardous road segments. The trend reflected in Figure 9 confirms this point. When the weights are relatively small, as they increase, the searched paths change accordingly, and the additional length cost tends to rise. However, when the weights are larger, increasing the weights results in significantly fewer changes to the obtained paths compared to when the weights are small, leading to more instances where the additional length cost remains unchanged.

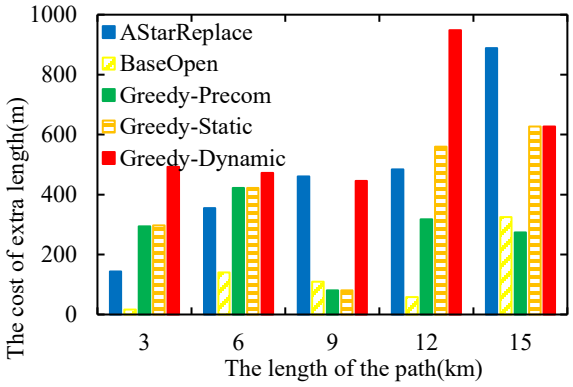


Fig. 10 Effects of the Euclidean distance between the start and end points on the extra length cost

Figure 10 shows the impact of the Euclidean distance between the origin and destination on the additional length cost in different algorithms. From the figure, it can be observed that when using the same algorithm to search for paths, a larger Euclidean distance between the origin and destination typically results in higher additional length costs. Among the three methods for estimating weights in the Greedy algorithm, Greedy-Dynamic usually incurs higher additional length costs. This is because the weights estimated by the other two methods, Greedy-Precom and Greedy-Static, typically do not fully utilize the travel length cost budget.

VI. Conclusion

Under the constraints of the road network, this paper proposes the problem of path planning in a spatio-temporal heatmap. Two query algorithms are designed, including an A* algorithm based on road segment hazard level filtering and an A* algorithm that adjusts the evaluation function using road segment heat levels. The weight w is introduced into the evaluation function to more effectively balance between the length of road segments and their safety or hazard attributes. Three methods are proposed to determine the value of the weight w in the evaluation function. Extensive experiments using real datasets have been conducted to verify the effectiveness of the algorithms.

Acknowledgment

The preferred spelling of the word “acknowledgment” in American English is without an “e” after the “g.” Use the singular heading even if you have many acknowledgments. Avoid expressions such as “One of us (S.B.A.) would like to thank ... .” Instead, write “F. A. Author thanks ... .” In most cases, sponsor and financial support acknowledgments are placed in the unnumbered footnote on the first page, not here.

REFERENCES

[1] Cao X, Chen L, Cong G, et al. Keyword-aware optimal route search[J]. Proceedings of the VLDB Endowment, 2012, 5(11): 1136-1147.

[2] Aljubayrin S, Qi J, Jensen C S, et al. Finding lowest-cost paths in settings with safe and preferred zones[J]. The VLDB Journal, 2017, 26(3): 373-397.

[3] Hu J, Yang B, Guo C, et al. Risk-aware path selection with time-varying, uncertain travel costs: a time series approach[J]. The VLDB Journal, 2018, 27(2): 179-200.

[4] Li Y, U L H, Yiu M L, et al. An experimental study on hub labeling based shortest path algorithms[J]. Proceedings of the VLDB Endowment, 2017, 11(4): 445-457.

[5] Fu A W C, Wu H, Cheng J, et al. Is-label: an independent-set based labeling scheme for point-to-point distance querying[J]. Proceedings of the VLDB Endowment, 2013, 6(6): 457-468.

[6] Thomsen J R, Yiu M L, Jensen C S. Effective caching of shortest paths for location-based services[C]//Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. 2012: 313-324.

[7] Ouyang D, Qin L, Chang L, et al. When hierarchy meets 2-hop-labeling: efficient shortest distance queries on road networks[C]//Proceedings of the 2018 International Conference on Management of Data. 2018: 709-724.

[8] Zhang M, Li L, Hua W, et al. Efficient batch processing of shortest path queries in road networks[C]//2019 20th IEEE International Conference on Mobile Data Management (MDM). IEEE, 2019: 100-105.

[9] Zhong R, Li G, Tan K L, et al. G-tree: an efficient and scalable index for spatial search on road networks[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(8): 2175-2189.

[10] Ouyang D, Yuan L, Qin L, et al. Efficient shortest path index maintenance on dynamic road networks with theoretical guarantees[J]. Proceedings of the VLDB Endowment, 2020, 13(5): 602-615.

[11] Papadias D, Shen Q, Tao Y, et al. Group nearest neighbor queries[C]//Proceedings 20th International Conference on Data Engineering. IEEE, 2004: 301-312.

[12] Hashem T, Hashem T, Ali M E, et al. Group trip planning queries in spatial databases[C]//International Symposium on Spatial and Temporal Databases. Springer, Berlin, Heidelberg, 2013: 259-276.

[13] Souffriau W, Vansteenwegen P, Berghe G V, et al. The planning of cycle trips in the province of east flanders[J]. Omega, 2011, 39(2): 209-213.

[14] Verbeeck C, Vansteenwegen P, Aghezzaf E H. An extension of the arc orienteering problem and its application to cycle trip planning[J]. Transportation Research Part E: Logistics and Transportation Review, 2014, 68: 64-78.

[15] Lu Y, Shahabi C. An arc orienteering algorithm to find the most scenic path on a large-scale road network[C]//Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems. 2015: 1-10.

[16] Chen C, Gao L, Xie X, et al. Enjoy the most beautiful scene now: a memetic algorithm to solve two-fold time-dependent arc orienteering problem[J]. Frontiers of Computer Science, 2020, 14(2): 364-377.

[17] Zeng Y, Chen X, Cao X, et al. Optimal route search with the coverage of users' preferences[C]//Twenty-Fourth International Joint Conference on Artificial Intelligence. 2015.

[18] Yao B, Tang M, Li F. Multi-approximate-keyword routing in GIS data[C]//Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. 2011: 201-210.

[19] Li W, Cao J, Guan J, et al. Retrieving routes of interest over road networks[C]//International Conference on Web-Age Information Management. Springer, Cham, 2016: 109-123.

[20] Pedersen S A, Yang B, Jensen C S. Anytime stochastic routing with hybrid learning[J]. Proceedings of the VLDB Endowment, 2020, 13(9): 1555-1567.

[21] Pedersen S A, Yang B, Jensen C S. Fast stochastic routing under time-varying uncertainty[J]. The VLDB Journal, 2020, 29(4): 819-839.

[22] He E Y, Boland N, Nemhauser G, et al. Dynamic discretization discovery algorithms for time-dependent shortest path problems[J]. INFORMS Journal on Computing, 2022, 34(2): 1086-1114.

Wang Yaxing(1997-), male, master. His research focuses on artificial intelligence and data mining.

Ma Xiaohui(1999-), female, master. Her research focuses on data query processing.

An Underwater Image Enhancement Method Based on Improved Dark Channel Prior Algorithm

Zhang Ruilin Sun Limei

School of Computer Science And Engineering, Shenyang Jianzhu University, Shenyang 110168 ,Liaoning, China

Abstract—Underwater images are usually affected by many factors such as ocean currents, light refraction, absorption and scattering of particles, and low light intensity, which lead to poor image quality. To solve these problems, an image fusion enhancement algorithm based on improved dark channel prior algorithm is proposed. This algorithm first aims at CIELUV color space to sharpen image edge features and enhance dark details, then applies multi-scale retinex with color restoration (MSRCR) to improve color distortion and enhance color saturation in RGB color space, and then use contrast limited adaptive histogram equalization (CLAHE).The fog effect is eliminated and the image contrast is enhanced. Finally, the fusion weight is calculated by the dark channel prior algorithm to linearly fuse multiple processed images. The experimental results show that the underwater images enhanced by this algorithm have obvious improvement in information entropy and peak signal-to-noise ratio (PSNR).

Keywords—Underwater image enhancement, Homomorphic filtering, Dark channel prior, Image fusion.

I. INTRODUCTION

The Marine environment is rich in biological, mineral and energy resources, which are of immeasurable value to the future survival and development of mankind. In order to effectively explore and utilize these resources, researchers have extensively used photoelectric imaging systems to identify and analyze underwater objects[1]. However, in shallow waters ranging from 0 to 200 meters, the complexity of Marine conditions presents many challenges, such as ocean currents, light refraction, the presence of micro-suspended particles, and low light intensity seriously affect image sharpness and detail. The forward scattering of underwater light will lead to blurred image details, and the backward scattering will lead to foggy image blur, which further reduces the image quality, and this problem is more prominent in turbidity waters. In addition, the rapid absorption of red light by water and deep penetration of blue light make underwater images often show a significant blue tone[2]. These factors jointly restrict the application of underwater images, such as object recognition[3], underwater bridge defect recognition[4] and target detection[5]. Therefore, the development and application of efficient image enhancement technology plays a crucial role in underwater environment exploration and resource utilization.

The core challenge of underwater image enhancement is to effectively acquire the original image and perform accurate feature extraction, especially for common color blurring and distortion problems[6]. In reference[7], aiming at the light distortion of water molecules and suspended particles, multi-scale convolutional feature extraction was proposed to build a network to increase image details and eliminate color bias. Reference[8] deeply analyzed the imaging principle of underwater images and the reasons for quality degradation, and reviewed the existing image and video enhancement methods. Reference[9] proposed effective methods to remove fog and restore color, aiming at the blurred, foggy and blue-green tones of underwater images affected by light absorption, scattering, color distortion and artificial light source noise; Reference [10] proposes a method of fusion of local contrast correction and multi-scale fusion, which solves low contrast and color distortion through red channel compensation and white balance; Reference [11] proposes an image enhancement method based on adaptive gamma value as a weight fusion of multiple algorithms. Neural network has a strong ability to extract features. With the rapid development of deep learning in recent years, many scholars try to integrate deep learning into it, and put forward some underwater image

enhancement methods based on deep learning. For example, reference [12] also integrates attention mechanism on the basis of neural network framework to extract more feature points. Then the color contrast and detail are further optimized by contrast learning. Reference[13] proposes a generative adversarial network that synthesizes multi-scale information and attention mechanisms. Different encoders are used to extract features, and multi-scale hybrid convolution is used to increase the adaptability of the network to different scale features. Reference [14] introduces the joint luminance and chrominance learning network (JLCL-Net), which avoids interference when separating luminance and chrominance of underwater images by decoupling key factors. Although deep learning-based approaches have achieved good results, they also face more challenges, such as large training data requirements, efficient network structure construction, and computational resources and time required for training models [15].

To sum up, aiming at the influence of flow and the strong scattering and attenuation of light by water in shallow sea environment, this paper proposes an advanced underwater image fusion and enhancement strategy combined with dark channel prior theory. Firstly, homomorphic filtering is used to balance the global illumination and reflection of the image. Then, applies the multi-scale retinex with color restoration (MSRCR)[16] to simulate the response of human eyes to brightness and contrast and make detailed adjustments. At the same time, the contrast limited adaptive histogram equalization (CLAHE)[17] method was used to further improve the local contrast and detail prominence of the image. Through the optimization of RGB color space, this strategy successfully corrects color deviation and significantly improves color saturation. Finally, dark channel prior technology[18] is used to combine the above methods to significantly improve image contrast and clarity, and effectively deal with the challenge of underwater environment on image quality.

II. Related work

II. I Color offset compensation processing

When applying dark channel technology, the compensation of color offset must be dealt with first. In the RGB three-color channel, red light has the longest wavelength, followed by green light and blue light is the shortest, so red light decays the fastest, resulting in underwater images often showing a significant blue tone. In order to improve color imbalance, the "gray world"[19] method is usually used for white balance processing to compensate for color loss and improve image appearance. However, due to the uneven color distribution of underwater images, the direct application of gray world method may lead to overcompensation of red channels and produce obvious red artifacts. Therefore, the red channel that precisely compensates the position of each pixel is particularly important. This compensation is calculated as shown in formula (1) :

$$I_{rc}(x) = I_r(x) + \alpha \cdot (\overline{I}_g - \overline{I}_r) \cdot (1 - \overline{I}_r(x)) \cdot I_g(x) \quad (1)$$

In the above formula, $I_r(x)$ and $I_g(x)$ represent the red and green channel values of pixel x positions in the image I, respectively. These channel values need to be normalized before processing to ensure that they are in the range [0,1]. $\overline{I}_g$ And $\overline{I}_r$ represents the average of the green and red channels in the image at all pixel positions, and these averages are used to adjust the color balance of the corresponding channels. In addition, the constant term α is used to further refine the degree of color correction, here set to 1. In addition to the red channel that needs to be compensated, the intensity of the blue channel will be significantly weakened under turbidity water conditions, making the overall image color more green, so the compensation for the blue channel cannot be ignored. The calculation method of the blue channel is similar to the compensation method of the red channel, and the specific calculation method is shown in formula (2) :

$$I_{bc}(x) = I_b(x) + \beta \cdot (\overline{I}_g - \overline{I}_b) \cdot (1 - \overline{I}_b(x)) \cdot I_g(x) \quad (2)$$

In formula (2), I_b represents the blue channel value of a specific pixel position x in the image I. In order to ensure the accuracy of processing, this channel also needs to be normalized before processing, limiting its value to a range of [0,1]. $\overline{I}_b$ represents the average value of all pixels of the blue channel I_b in the entire image, and the calculation of this average value is crucial for balancing and correcting the blue channel. β As a constant term, also set to 1 with formula 1, it is used to fine-tune the compensation process in order to recover the true color of the image more precisely. In this way, even under cloudy water conditions, the attenuation of the blue channel can be effectively compensated, thereby improving the color performance of the entire image. After compensating the attenuation of red and blue channels, the overall color deviation of the image is estimated and compensated by the grayscale world algorithm. However,

even if this method can effectively correct the color, the image may still face the problem of blurred details and insufficient contrast, so in addition to color correction, the image needs to be further processed to enhance the image clarity and contrast, so as to show the underwater details more clearly.

This is intended as an authoring template, not a final production template. It is not intended to match the final published format. Page count in the template is an estimate. Do not adjust line and character spacing to fit your paper to a specific length.

II.II Homomorphic filtering

Due to the absorption and scattering of light as it travels through water, the underwater image presents uneven illumination, resulting in blurred image details. The homomorphic filtering algorithm[20] can compress the brightness range of the image and enhance the image contrast. Image $f(x,y)$ can be represented by the product of its illumination function $i(x,y)$ and its reflection function $r(x,y)$, as shown in formula (3) :

$$f(x,y)=i(x,y)r(x,y) \tag{3}$$

Where $i(x,y)$ describes the lighting properties of the image and satisfies $0<i(x,y)<\infty$, which is independent of the detailed properties of the image. Because the relative change in illumination is small, it can be considered a low-frequency component of the image. The corresponding is $r(x,y)$, which describes the reflection intensity of the image and satisfies $0<\mathrm{r}(x,y)<1$. Because different objects reflect light differently and have mutability, they can be regarded as high-frequency components of the image. By dealing with the effects of illumination and reflection on the gray value of the image, the detailed features of the shadow region can be obtained. After logarithmic operation on formula (3) and Fourier transform, formula (4) can be obtained:

$$F[\ln f(x,y)] = F[\ln i(x,y)]+F[\ln r(x,y)] \tag{4}$$

By rewriting formula (4), formula (5) can be obtained:

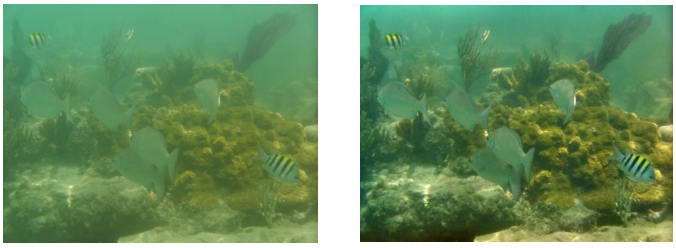
$$\begin{aligned} s(x,y)&=F^{-1}[S(x,y)]=F^{-1}[H(x,y)F(x,y)]= \\ &F^{-1}[H(x,y)i(x,y)]+F^{-1}[H(x,y)r(x,y)] \end{aligned} \tag{5}$$

Where $H(x,y)$ is a homomorphic filter function, which can be applied to the lighting component and the reflection component respectively. The filter adopts a Gauss high-pass filter, and the function satisfies formula (6) :

$$H(x,y)=(\gamma_H-\gamma_L)\left(1-e^{-cD^2(x,y)/D_0^2}\right)+\gamma_L \tag{6}$$

in formula (6): $D^2(x,y)=x^2+y^2$, γ_L and γ_H are range parameters used to control the amplitude of the filter, which satisfy $\gamma_L<1$ and $\gamma_H>1$. The constant c is used to control the sharpness of the slope transition between γ_L and γ_H, that is the steepness of the low frequency to high frequency transition segment.

To ensure that the color of the image remains unchanged after color deviation correction, the image is converted to LUV color space processing[21]. In this space, U and V color components are kept unchanged, and only the brightness component L is homomorphically filtered to enhance image contrast. After homomorphic filtering, the brightness component L is converted back to the RGB color space together with the color components U and V. This solves the problem that the dark feature of underwater image is not obvious due to the uneven brightness. The comparison results are shown in Figure 1.



(a) original image (b) homomorphic filtering

Fig.1 Comparison of the original image after homomorphic filtering in the LUV color space

III. METHODOLOGY

III.I MSRCR

The multi-scale retinex with color recovery algorithm can eliminate light interference and obtain the inherent reflection characteristics of the target object, the expression of which is shown in formula (7) :

$$I(x,\ y)=L(x,\ y)R(x,\ y) \tag{7}$$

The expression is similar to that of homomorphic filtering algorithm, but the meaning is different. In this formula, assuming that the initial image is $I(x,\ y)$, then $L(x,\ y)$ is the incident component and $R(x,\ y)$ is the reflected component. The multi-scale retinal contrast enhancement technique with color recovery can yield the formula (8):

$$\begin{cases} R_{MSRCRi}(x,y) = C_i(x,y) * R_{MSRi}(x,y) \\ C_i(x,y) = \eta\left(\log(\lambda \cdot I_i(x,y)) - \log\left(\sum_{i=1}^{N} I_i(x,y)\right)\right) \\ R_{MSRi}(x,y) = \sum_{n=1}^{n_s} \mu_n \left(\log(I(x,y)) - \log(I(x,y) \cdot G_n(x,y))\right) \end{cases} \quad (8)$$

Where, $R_{MSRi}(x, y)$ represents the high frequency detail image obtained by multi-scale filtering, and $G_n(x, y)$ is the result of single-scale Gaussian filtering. $\mu_n = \frac{1}{3}$ represents the scale parameter and its weight, and $R_{MSRCRi}(x, y)$ is the high-frequency detail image of the i channel multi-scale filter combined with the color recovery factor. η and λ are nonlinear intensity control factors and information parameters, respectively.

III.II CLAHE

In order to enhance the edge information of the image as much as possible and improve the image quality, histogram equalization algorithm is adopted after the processing of the above method. Meanwhile, in order to avoid the interference of noise, this paper uses the contrast limited adaptive histogram equalization (CLAHE). Contrast limits are designed to cut pixels in a histogram that exceed a threshold and distribute those portions evenly across the histogram, limiting their amplitude. The calculation method of limiting threshold C is shown in formula (9) :

$$C = \frac{N}{L} + \sigma\left(\frac{N - \frac{N}{L}}{L}\right) \quad (9)$$

Where, N is the total pixels in a certain area, L is the maximum grayscale sequence in the area, and σ is the truncation coefficient, ranging between [0,1).

III.III Dark channel prior algorithm

Dark channel prior algorithm was first proposed by He et al[22] and is a classic fog removal algorithm. After a large number of image experiments, He et al. concluded that in most non-sky local areas, the pixel value of at least one color channel (RGB) is low, while in non-sky areas, the prior image intensity value of the dark channel of the non-fog image is lower than the corresponding value of the fog image. Atmospheric fog images are very similar to underwater fog images, both of which have problems such as color distortion, low contrast and distant blur. Therefore, the dark channel prior algorithm can also be applied to underwater image enhancement. In this paper, the atmospheric scattering model in the dark channel algorithm is used to model underwater images. Dark channel prior theory is applied to underwater images to generate underwater dark channel prior images. The calculation method is shown as formula (10) :

$$I_{\text{dark}}(x) = \min_{x \in \Omega(x)} \left(\min_{c \in \{RGB\}} I_c(x)\right) \quad (10)$$

In the above formula, $I_{\text{dark}}(x)$ is the prior image of the dark channel, which satisfies $I_{\text{dark}}(x) \to 0$. That c is a channel in RGB and $I_c(x)$ is x pixel value in a channel in the underwater image, where is a local window centered by a pixel, and the window size is set to 15×15.

III.IV Improved dark channel prior algorithm

By analyzing the results of homomorphic filtering, MSRCR and CLAHE algorithms for underwater image enhancement, the following conclusions can be drawn: Homomorphic filtering algorithm can alleviate the uneven brightness of images and improve the characteristics of dark parts. MSRCR algorithm can effectively improve brightness and color saturation, CLAHE algorithm has a certain fog removal effect. The above algorithms have their own advantages and can play their own advantages in different application scenarios. However, due to the complexity of underwater environment, in order to solve the challenges brought by underwater environment images as much as possible, only one algorithm cannot achieve satisfactory results, which requires the integration of multiple image enhancement algorithms to jointly solve the problem of underwater image quality reduction. Therefore, this paper proposes to calculate the image contrast through the dark channel prior algorithm, and then fuse the above three algorithms by weighted contrast ratio of the three groups of images. Through this method, the advantages of the above algorithms can be fused to further improve the robustness of the underwater image enhancement algorithm. The specific calculation method is to first calculate the dark channel prior weight coefficient ω_{DCP} of the image, which is calculated from the exp function of the mean value of the dark channel prior image. The specific calculation process is shown in formula (11) :

$$\omega_{DCP} = \exp\left(-\frac{I_{mdark}}{v^2}\right) \quad (11)$$

Where I_{mdark} is the mean of the dark channel prior image and v is a constant parameter. Because the smaller the result value obtained by the dark channel prior algorithm, the less fuzzy fog in the original image, the higher the image contrast, the larger the result value obtained, the lower the contrast of the original image, and the two are inversely proportional, it is necessary to take a negative sign for the exp function. The empirical number[22] can be obtained by calculating the mean value of the dark channel prior image of the underwater image. $v=10$ can effectively ensure that the dark channel prior weight coefficient is not too small, thus improving the computational efficiency. The final fusion weight coefficient can be calculated according to the prior weight coefficient of the dark channel, and the formula (12) can be obtained:

$$W_i = 1 - \frac{w_{DCP_i}}{w_{DCP_{other}} + w_{DCP_{another}}} \tag{12}$$

Where, W_i is the final fusion weight, ω_{DCPi} is the dark channel prior weight coefficient of the current image, and $w_{DCP_{other}}$ and $w_{DCP_{another}}$ are the dark channel prior weight coefficients of other images. The improved underwater image enhancement fusion algorithm of the dark channel can be expressed as follows:

Step 1: The image is transferred to LUV space, and the L brightness component is converted back to RGB space after homomorphic filtering algorithm.

Step 2: The R, G and B channels of the image processed by the homomorphic filtering algorithm are fused with the R, G and B channels of the image processed by the MSRCR algorithm respectively.

Step 3: Calculate the dark channel prior weight coefficient ω_{DCP} of the image to be fused according to formula (11), and then use formula (12) to calculate the weight coefficient W_i of the second fusion.

Step 4: Fuse the homomorphic filtered image with the MSRCR processed image according to the weight coefficient, and then fuse the first fused image with the CLAHE processed image again.

Step 5: Merge the three RGB fusion channels and complete the secondary fusion to get the final output image. The complete structure of underwater image enhancement fusion algorithm is shown in Figure 2.

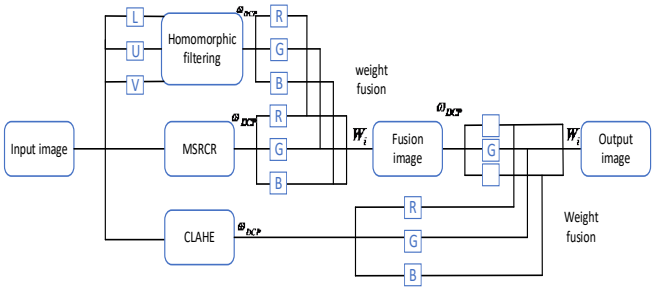


Fig. 2 Overall architecture of underwater image enhancement fusion algorithm

IV Experiments

In order to prove the effectiveness of the proposed algorithm, the experiment is conducted based on the Underwater Image Enhancement Benchmark (UIEB) dataset[23], which uses images from underwater images searched on the Internet, including 950 real-world underwater images, 890 of which have good recovery results. Failure to obtain satisfactory reference images for 60 images is considered challenging data. The experimental system environment is *I*ntel(R) Core(TM) i5-7300HQ @2.50Ghz , the experimental algorithm is carried out in Matlab R2020b environment. In the experiment, five representative images were selected from 890 images, which have different resolutions and contain different scenes, different objects and different tones. The final experimental results were compared with the algorithms in the four literatures, and the superiority of the proposed algorithm was proved through subjective human visual effect and objective numerical evaluation.

Figure 3 shows the effect of five images enhanced by different algorithms. Dark channel and CLAHE algorithm improved the clarity and naturalness of the images, but failed to eliminate color bias, and the images processed by dark channel algorithm became significantly darker. MSRCR algorithm significantly improves image brightness and color saturation, but color deviation still exists and details are blurred. The algorithm proposed in this paper effectively improves color deviation, improves sharpness and contrast, and enhances dark details to provide excellent visual effects.

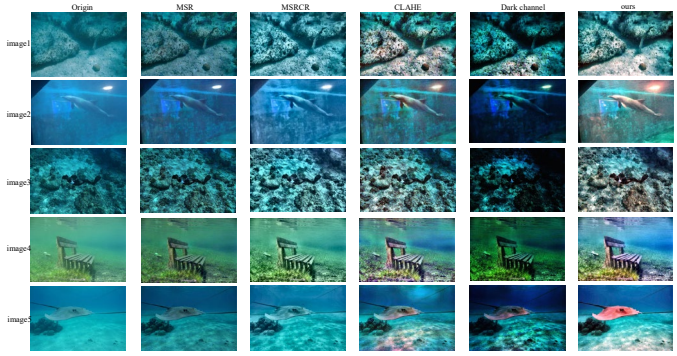


Fig 3 Comparison of experimental results of different algorithms

Compared with the subjective evaluation of human eyes, objective image quality is more convincing. Therefore, the following three objective image quality evaluation indicators are also adopted in this paper to measure the effect of the algorithm: information entropy(IE)[24], peak signal-to-noise ratio(PSNR)[25]and Naturalness Image Quality Evaluator(NIQE)[26].The information entropy reflects the amount of information contained in the image, and the larger the value, the more useful information is contained in the processed result, and the better the image recovery quality is. PSNR is the logarithm of the ratio between the square of the maximum signal value and the mean square error before and after image processing. It has a wide range of applications in image processing. The larger the value, the smaller the distortion, that is, the higher the quality of the image after processing. However, in some cases, the higher the value of PSNR, the better, and the texture details after some image processing are not in line with the visual experience of human eyes. Therefore, this paper also adopts the non-reference image evaluation index NIQE, which is more in line with the visual habits of human eyes. A lower value means a higher naturalness of the image and a more natural visual effect. On the contrary, a higher value indicates a low degree of naturalness, and there may be unnatural or distorted phenomena.

Table 1 to Table 5 respectively show the objective evaluation data values calculated by MSR, MSRCR, CLAHE, dark channel algorithm and the improved dark channel algorithm

Table 1 Comparison of objective evaluation indicators

	IE	PSNR/dB	NIQE
MSR	7.8978	29.6266	3.7731
MSRCR	7.3096	24.9421	3.5853
CLAHE	7.7244	29.4715	3.5636
Dark channel	7.3797	25.8106	3.6761
ours	8.4759	30.8448	3.0347

Table 2 Comparison of objective evaluation indicators

	IE	PSNR/dB	NIQE
MSR	7.7414	29.4218	4.5622
MSRCR	6.5704	24.0765	5.1314
CLAHE	7.7109	29.2300	4.4891
Dark channel	7.3497	25.1787	5.0604
ours	8.2841	30.0773	4.1767

Table 3 Comparison of objective evaluation indicators

	IE	PSNR/dB	NIQE
MSR	7.8650	34.8769	2.9798
MSRCR	5.9688	24.1143	3.1151
CLAHE	7.7755	34.3390	2.8434
Dark channel	7.5353	32.1111	2.8034
ours	8.3972	35.0414	2.6283

Table 4 Comparison of objective evaluation indicators

	IE	PSNR/dB	NIQE
MSR	7.6080	26.5105	2.9990
MSRCR	6.9095	24.0698	2.6765
CLAHE	7.6084	29.1125	2.8561
Dark channel	7.0816	24.0691	2.9041
ours	8.2923	34.2372	2.5742

Table 5 Comparison of objective evaluation indicators

	IE	PSNR/dB	NIQE
MSR	7.6912	27.2499	5.7278
MSRCR	7.1807	24.3587	6.2540
CLAHE	7.2504	29.5911	5.6247
Dark channel	7.1194	24.1267	6.6918
ours	8.8312	34.5418	5.5544

proposed in this paper. First of all, the information entropy index reflects the amount of information in image restoration, and higher information entropy indicates that the processing results contain more useful information. On the above five images, the proposed algorithm has the highest information entropy, indicating that the images enhanced by the proposed algorithm contain more useful information. Taking Image 1 in Table 1 as an example, compared with the other four algorithms, the proposed algorithm has improved by 7.3%, 15.9%, 9.7% and 14.9% respectively, which proves the excellent performance of the proposed algorithm in image enhancement.

Second, PSNR is used to measure the degree of distortion in the processing result, and a higher PSNR value usually means a smaller distortion. The algorithm in this paper presents the highest PSNR value on all five images. Similarly,

taking Image 1 in Table 1 as an example, the algorithm in this paper has improved by 4.1%, 23.6%, 4.6% and 19.4% respectively compared with the other four algorithms. This data shows that the proposed algorithm can significantly reduce image distortion in the process of image enhancement, and can retain image details and structure information more effectively to achieve higher quality image enhancement.

Finally, this paper uses the NIQE index to evaluate the naturalness of the image, which is an objective and subjective evaluation. A lower NIQE value indicates that the processed image is more natural. The algorithm in this paper presents the lowest NIQE value on all five images. Similarly, taking Image 1 in Table 1 as an example, the algorithm in this paper reduces 19.6%, 15.3%, 14.8% and 17.4%, respectively, compared with the other four algorithms. This data shows that the image enhanced by the proposed algorithm has higher image naturalness than the other four algorithms and is more in line with human vision

In summary, according to objective evaluation indicators, the improved dark channel prior algorithm has excellent performance in terms of information entropy, PSNR and NIQE. Compared with the other four algorithms, this algorithm performs better in image enhancement, distortion reduction and naturalness enhancement. Through the research results, it can be concluded that the improved dark channel prior algorithm proposed in this paper has more potential than other underwater image enhancement algorithms, which can improve the quality and naturalness of image enhancement, and is also more in line with the visual habits of human eyes.

V. Conclusion

Aiming at the problems of fuzzy details, contrast reduction and color distortion in underwater images due to the complexity and variability of shallow sea environment, a weighted fusion underwater image enhancement method based on improved dark channel prior algorithm is proposed in this paper. The color distortion caused by the selective absorption of light is effectively corrected by compensating the loss of red and blue channels. By homomorphic filtering the L component of the corrected image in LUV space, the fuzzy and dark detail problems caused by forward light scattering and non-uniform illumination are solved. The problem of underwater image atomization caused by light backscattering is solved by using CLAHE algorithm in RGB space. By using the MSRCR algorithm in RGB space to solve the brightness problem of underwater images, color saturation and overall contrast are improved. The final experimental results show that the proposed algorithm is superior to the other four algorithms in terms of subjective visual effect and has richer color contrast. Objectively, the proposed algorithm also achieves better results than the other four algorithms in terms of NIQE, information entropy and PSNR, especially for dark images.

Acknowledgment

This work is supported in part by Science and technology project of Liaoning Provincial Archives Bureau (2023-X-050).

References

[1] Zhong Jian Dan, Key Technique of Object Recognition and Detection in Optoelectronic Imaging.[D]. University of Electronic Science and Technology of China, 2018.

[2] Yan Zheping, Qu Siyu, Xing Wen. An overview of underwater image enhancement methods[J]. CAAI Transactions on Intelligent Systems, 2022, 17(5):860-873.

[3] Yang Wenjing, Chen Ming, Feng Guofu. Fish recognition method for underwater video based on image enhancement[J]. Laser and Optoelectronics Progress, 2021, 58(22): 294-303.

[4] Li Zerong, Liu Airong, Chen Bingcong et al. Bridge underwater structural defects detection based on fusion image enhancement and improved YOLOv7[J/OL]. Engineering Mechanics, 1-9[2024-03-21]. http://kns.cnki.net/kcms/detail/11.2595.o3.20240304.1525.032.html.

[5] Zhang Shang, Li Mengsi, Chen Yonglin et al. An improved YOLOv7 based algorithm for ship target detection in SAR images[J]. Electronics Optics and Control, 2024, 31(05): 46-53.

[6] Al-Dwairi M O, Alqadi Z A, Abujazar A A, et al. Optimized true-color image processing[J]. World Applied Sciences Journal, 2010, 8(10): 1175-1182.

[7] Liu Yuzhen, Liu Meiyi, Lin Sen et al. Underwater image enhancement based on multi-scale feature fusion and attention network[J]. Journal of Computer-Aided Design and Computer Graphics, 2023, 35(05): 685-695.

[8] Yu Haifeng. Research on underwater image quality enhancement and object detection algorithm[D]. Yanshan University, 2022.

[9] Pan Dongyu. Research and Application of Degraded Image Restoration Methods in Underwater and Foggy Environment[D]. Jilin University, 2023.

[10] Li Huakun, Zhao Lei, Li Heng et al. Underwater image

enhancement with multi-scale fusion and detail enhancement.[J/OL]. Mechanical Science and Technology for Aerospace Engineering, 1-9[2024-03-21].https://doi.org/10.13433/j.cnki.1003-8728.20230014.

[11] Li Guanghao, Xi Zhihong. Image Enhancement Algorithm of Underwater Uneven Illumination Based on Image Fusion[J]. Computer Simulation, 2023, 40(04): 330-335.

[12] Wang Yue, Fan Huijie, Liu Shiben, et al. Underwater Image Enhancement Based on Multi-Scale Attention and Contrast Learning[J]. Laser and Optoelectronics Progress, 2024, 61(04): 559-567.

[13] Xia Xiaohua, Zhong Yuquan, Hu Peng, et al. Underwater image enhancement synthesizing multi-scale information and attention mechanisms[J]. Optics and Precision Engineering, 2024, 32(10): 1582-1594.

[14] Hao Zhenhua. Underwater Image Enhancement via Multi-color Space[D]. Dalian University of Technology, 2022.

[15] Zhang Tianchi, Liu Yuxuan. Research Progress of Underwater Image Processing Based on Deep Learning[J]. Computer Science, 2024, 51(S1): 283-294.

[16] Wang Jinbao, Lu Ke, Xue Jian, et al. Single image dehazing based on the physical model and MSRCR algorithm[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2017, 28(9): 2190-2199.

[17] Reza A M. Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement[J]. Journal of VLSI signal processing systems for signal, image and video technology, 2004, 38: 35-44.

[18] Xu Haoran, Guo Jianming, Liu Qing, et al. Fast image dehazing using improved dark channel prior [C]// IEEE international conference on information science and technology. IEEE, 2012: 663-667.

[19] Wan Mustapa W A, Yazid H. Conversion of the retinal image using gray world technique[J]. Journal of Biomimetics, Biomaterials and Biomedical Engineering, 2018, 36: 70-77.

[20] Seow M J, Asari V K. Ratio rule and homomorphic filter for enhancement of digital colour image[J]. Neurocomputing, 2006, 69(7-9): 954-958.

[21] Kekre H B, Mishra D, Saboo R S. Comparison of image fusion techniques in RGB & Kekre's LUV color space [C]// 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE). IEEE, 2015: 114-120.

[22] He Kaiming, Sun Jian, Tang Xiaoou. Single image haze removal using dark channel prior[J]. IEEE transactions on pattern analysis and machine intelligence, 2010, 33(12): 2341-2353.

[23] Li Chongyi, Guo Chunle, Ren Wenqi, et al. An underwater image enhancement benchmark dataset and beyond[J]. IEEE transactions on image processing, 2019, 29: 4376-4389.

[24] Tsai D Y, Lee Y, Matsuyama E. Information entropy measure for evaluation of image quality[J]. Journal of digital imaging, 2008, 21: 338-347.

[25] Wang Yuanji, Li Jianhua, Lu Yi, et al. Image quality evaluation based on image weighted separating block peak signal to noise ratio [C]// International Conference on Neural Networks and Signal Processing, 2003. Proceedings of the 2003. IEEE, 2003, 2: 994-997.

[26] Liu Yutao, Gu Ke, Zhang Yongbing, et al. Unsupervised blind image quality evaluation via statistical measurements of structure, naturalness, and perception[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2019, 30(4): 929-943.

Zhang Ruilin(1998-), male. His research focuses on image enhancement and image stitching.

Sun Limei, female, PhD, associate professor. Her research focuses on artificial intelligence for image enhancement.

Call for Papers: Journal of AI Applications

[Journal Introduction]

The Journal of AI Applications, with ISSN 3021-3215 (Print) and ISSN 3021-3223 (Online), is published under the sponsorship of the Global Academic Research Association (GARA) in New Zealand and approved as a serial publication by the National Library of New Zealand. This international academic journal focuses on the latest research findings and practical applications in the field of artificial intelligence, aiming to provide a high-end platform for global researchers, engineers, and industry experts to exchange ideas and share innovative achievements. The journal emphasizes the latest advancements in AI technology, particularly its practical applications and impacts across various industries, fostering interdisciplinary collaboration and promoting the sustainable development of AI technology.

[Columns]

To better meet the needs of authors and readers, this issue of the *Journal of AI Applications* features the following three key columns and invites experts, scholars, and researchers in related fields to submit their manuscripts:

1. AI Algorithms: This column welcomes research papers on the innovation, optimization, and improvement of AI algorithms. It includes but is not limited to the latest algorithm design, performance evaluation, and theoretical analysis in areas such as deep learning, reinforcement learning, transfer learning, graph neural networks, natural language processing, and computer vision. Authors are encouraged to explore the innovation points of algorithms, application scenarios, and advantages compared to existing methods.
2. AI Models: This column focuses on the construction and optimization of AI models, covering model architecture design, parameter tuning, training strategies, and model compression. We look forward to receiving research outcomes that significantly improve model performance, reduce computational costs, or enhance model interpretability. Research explorations on cutting-edge directions such as pre-trained models, generative models, and multimodal models are also welcome.
3. AI Application Cases: This column aims to showcase practical application cases of AI technology in various industries, including but not limited to smart manufacturing, smart cities, healthcare, finance tech, education, entertainment, and more. Authors are encouraged to share the lessons learned from successful cases, technical challenges and solutions, as well as specific examples of how AI technology drives industrial upgrading and social progress.

[Submission Requirements]

1. Originality: Submitted papers must be unpublished original works and are strictly prohibited from being submitted to multiple journals simultaneously.

2. Formatting Standards: Please prepare manuscripts according to the submission guidelines of the Journal of AI Applications, including abstract, keywords, main text, conclusion, references, etc.
3. Language Requirements: Papers should be written in English with correct grammar and clear expression. Translation services are provided for Chinese manuscripts.
4. Clear Charts: All charts must be clear and accompanied by detailed explanations.
5. Copyright Agreement: Authors are required to sign a copyright transfer agreement as per the requirements.
6. Ethical Statement: Papers involving human subjects or animal experiments must be accompanied by relevant ethical review certificates.

[Submission Method]

Please send the electronic version of the manuscript (in Word or PDF format) to the submission email: ai_application@163.com(China), info@gara.org.nz. The email subject should indicate "Submission - Column Name - Author's Name".

[Fees]

There are no review or publication fees for this issue. A translation fee of RMB 200 Yuan per 1000 Chinese characters will be charged for Chinese manuscripts.

[Contact Information]

For any questions or further information about submission details, please contact us via email or visit the "Contact Us" page on the official website of the *Journal of AI Applications* (http://www.gara.org.nz).

The editorial department of the *Journal of AI Applications* looks forward to your excellent submissions and working together to promote the prosperity and development of AI technology!

ISSN 3021-3215（Print），3021-3223（Online）