GeoVis: A Data-driven Geographic Visual Recommendation System via Latent Space Encoding

Category: Research

Paper Type: algorithm/technique

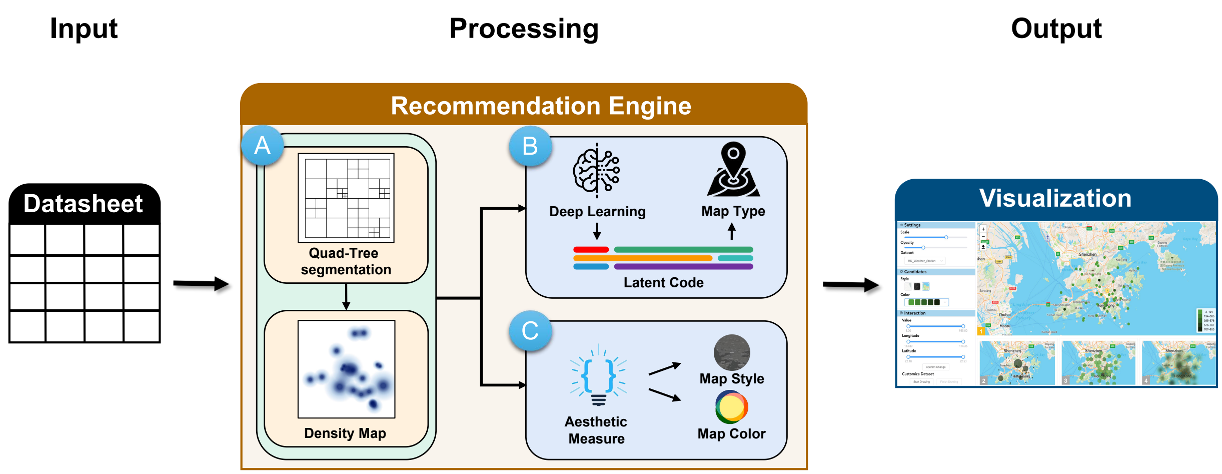

Fig. 1. This figure shows the overall pipeline of GeoVis. In the recommendation engine part, (a) the input datasheet is divided into quad-tree, and the adaptive bandwidth is calculated. Then, according to the calculation results, the kernel density is estimated and the density map is generated. On this basis, (b) we use the deep learning model to encode the density map, and finally recommend a variety of visual charts through latent code. At the same time, (c) GeoVis calculate the candidates of map style and color based on aesthetic measure and present them together with the recommended map type to improve the quality of visualization results. (d) Users can browse the recommendation results in the system for data analysis.

Abstract—As one of the effective means of representing geographic information, geographic visualization can directly improve the cognitive efficiency of users who are perceiving geospatial data. The existing geographic information visualization relies heavily on the background knowledge and visualization skills the data workers own. Therefore, the geographic visualization task is usually very time-consuming and challenging. To lower the threshold of visualization of geographical data, we propose a novel recommendation system of geographic information visualization called GeoVis. This system extracts the distribution characteristics with adaptive kernel density estimation and recommends the map type that can best reflect the regularity of data distribution based on latent code. The key idea of how the data-driven recommendation work is to use latent code to express and decouple data features and then learn the mapping between data features and visual styles. At the same time, this system recommends design choices (e.g. map styles and color schemes). Users only need to browse the recommendation results to realize explorations and analyses of the dataset, which will greatly improve their work efficiency. We conduct a series of evaluation experiments on the proposed system, including case study and user research. The experiment results show that the system is practical and effective, and can perform the task of recommending informative and aesthetic geographical visualization results well.

Index Terms—Geographic Information, Visualization Recommendation, Deep Learning, Density map, Latent Space.

1 Introduction

In the process of urban intelligence [1, 3], people's working styles and lifestyles are undergoing profound changes, and people's activities are closely related to spatial information [19]. How to use advanced information technology to process geospatial data and to maximize the efficiency of the city and to achieve the best quality of life are the keys to the realization of intelligent city [4]. An increasing amount of equipment can record and sense a large amount of geospatial data, such as the spatial flow of people, the trajectories of vehicles and the distribution of shared bicycles. These data have different varieties and granularities. Understanding these data with location labels is important for urban management, helping to solve urgent problems such as urban traffic and public resource scheduling in real time and to provide strong support while making scientific and accurate decisions.

Visualization of geospatial data is one of the important ways to understand these data. It presents geographic information to users in a visual way, which can help users, who can gain deeper cognition afterwards, to quickly grasp the trends and regularity of information change, greatly reducing the difficulty of understanding data. In this way, the efficiency of the data analysis could be improved [49]. Because the fact that visualization can clearly display a variety of data, and it can guide users to analyze and infer effective information from the results of the visualization, as early as the 16th century, people drew the first traffic map in history to show the traffic situation of Rome [36].

However, according to our research, due to the characteristics of multi-dimensional, complex hierarchy and large amount of data of geospatial data, users always choose visualization styles through some human-defined attributes [59]. Although this artificial filtering method is useful, it requires users to understand the specific details of tasks in advance and to possess good visualization skills. Obviously, this is not an easy condition for a user who doesn't know much about data processing or has little visualization experience. Therefore, how to better help users understand the essential characteristics of geospatial data in a more intuitive way, accurately recommend visual chart types to users automatically, and use more beautiful visualization styles to achieve excellent perception effect are current research hotspots [5].

Data oriented automated chart recommendation has been widely studied and developed in recent years [52]. For example, Deepeye [30, 41] supports the introduction of rules of domain experts to sort the visualization results. Vizml [23] learns visual design choices from a large number of dataset-visualization pairs. It is not difficult to see that automatic recommendation of charts based on data can effectively reduce the threshold of data analysis and improve the efficiency of data analysis. Inspired by these recommendations, we hope to develop an automated recommendation system of visual charts for geospatial data, in order to solve the current challenges of geospatial data visualization. However, developing such a system faces three serious challenges:

Geospatial data feature mining. Generally, it is in the habit of people to use dots to mark maps, which can display effective information in limited space. However, with increasing amounts of data, an increasing number of points will overlap in some areas, which will make the user unable to extract effective information accurately. How to effectively mine the geospatial features in complex datasets and extract the changing trend and law of information [25] have put forward higher requirements for our system.

Recommendations for geographic visualization chart types. With the development of technology, corresponding graphic recommendation systems have emerged to help nondata scientists discover sustainable insights. However, most of the current visualization automatic recommendation systems only consider the numerical characteristics of data but ignore the location attributes of geospatial data. Besides, there are many different design strategies for geospatial data. How to recommend the most appropriate charts for different data sets is a difficult problem.

Reasonable visual coding. Human attention is very limited, and effective visual coding can directly affect users' information perception and cognitive ability [9]. For the recommendation results of visual chart types, its expressiveness and visual aesthetics depend on whether a reasonable visual coding is selected. For the candidate visual coding, our system needs an effective evaluation algorithm to determine the best visual coding, so that users can quickly analyze and make decisions on the visual results.

In order to solve the problems raised, this paper designs an automatic recommendation system named **GeoVis**, which can recommend the most appropriate chart types for different datasets, and effectively help users intuitively perceive the information contained in the data. Firstly, the latent code corresponding to each dataset is extracted, and the chart type is automatically selected for the original geospatial data. Then, due to the strong dispersion of geospatial data, we use quad-tree algorithm for adaptive kernel density estimation to better describe the spatial distribution law and obtain the estimation results. Secondly, the density map of kernel density estimation is encoded by a trained encoder. Then, we use softmax classifier to sort the dimensionality reduction results, and realize the recommendation of visual charts of original geospatial data. Finally, we carry out the corresponding visual presentation on the front end and supported a variety of visual effects. To verify the use effect of GeoVis, we compare our recommended model with other models, and apply the system to different datasets. The results show that the visualization effect have achieved excellent results.

The contributions of this paper are as follows:

1. **System:** we propose an automatic recommendation method for different charts under the given scattered data, so that users can easily visualize the local geographic information data. As far as we know, this is the first system to automatically recommend charts for geographic information data.

2. **Model:** we introduce a method to extract distribution features of a given geographic data, and propose a recommendation model based on latent code to learn the mapping relationship between data features and map type. We find that the recommendation model by latent code has high accuracy. At the same time, we use aesthetic measure to recommend design choices.

3. **Evaluation:** We conduct a user survey to verify the consistency between the charts generated by our system and users' visual perception. Besides, our system also provides interactive functions, which improves the practicability of our system.

2 Related Work

Data visualization is an important means for users to understand the connotation of data and the law of data hiding. Information can be transmitted efficiently through graphical symbols. Our related work is divided into 2.1 Automatic Visualization Recommender Systems and 2.2 Geographic Information Visualization.

2.1 Automatic Visualization Recommender Systems

Visualization recommendation system [50] is an important way to help users explore data, it has novel operation interface and convenient user interaction design, which greatly decrease the technical difficulty of visualization and improve the understanding of abstract data [42].

Early studies were based on rules, including Voyager [53], DIVE [24], Show me [32]. Although their visualizations look impressive and artistic, these methods are greatly dependent on the guidelines made by them, which has obvious limitations considering visualization design space will have explosive growth when data dimension increases. Similarly, the task oriented recommendation system [46, 48] requires not only clear tasks, but also corresponding detailed rules, so that data personnel can carry out corresponding visualization based on tasks [37]. This may require a lot of time and energy. The data-driven method can reveal the insight of the data itself more accurately and directly, and has a wider range of application scenarios, which can find hidden rules and knowledge from the data [26] well.

Most of the recent researches are ML-based Visualization Recommender Systems [51, 54], extracting the data features with automatic ways to select the corresponding chart visualization, so users are free to choose and search. The ML-based model can achieve efficient visual design and generation and has a relatively low threshold for inexperienced analysts in the process of exploring the laws of big data [56]. Even novices can be provided with sufficient guidance in the process of visual analysis through rich interactive responses, and can also find hidden laws from massive data.

Deepeye [30, 41] uses a decision tree for classification visualization and trains a supervised learning-to-rank model. Vizml [23] uses the aggregation function to map the dataset to 841 features, then analyzes the design choices in each visualization. These features are used to predict users' visual choices. Data2vis [15] describes visualization generation as a language translation problem and uses LSTM to learn an end-to-end model for automatically generating visualization effects from given datasets. Qian et al. [40] used the ML based model to build the evaluation framework by scoring the attribute combinations and visual configurations, and proved the effectiveness of the ML based visual recommendation system through user experiments. KG4Vis [28] uses the method of constructing a knowledge graph to mine data features, data columns and visual design, and select the relationship between three entities to realize visual recommendation. Further, by combining multiple charts to generate a dashboard [31, 55], users can explore data characteristics from different angles at the same time.

These methods reduce the technical difficulty of users as much as possible, and effectively reduce the burden and constraints of users in creating charts, but still maintain a good visualization effect, and can reveal the hidden data characteristics and change trends in the dataset. Inspired by these methods, GeoVis adopts a method based on deep learning to automatically recommend chart types that can reveal the distribution characteristics of geographic data to the greatest extent.

2.2 Visualization of Geographic Information

Geospatial data is a kind of data with geographic location information [6]. To show the rules of spatial distribution more succinctly and intuitively, and at the same time mining deeper information, it is a common research method to visualize the geospatial data in the way of map [22].

It is in accordance with people's habit to see maps by using dots to mark them [12]. It can display effective information in a limited space. A typical example [43] is to use the scatter diagram to describe the distribution of races in Chicago in 2000. The dots of different colors represent different races. It can be clearly found that different races live together in the places where the races meet.

However, when there are a large number of overlapped points in the data, the user cannot extract the information accurately [44], especially about the distribution and density of data [58]. Therefore, some scholars have proposed a series of methods to solve the overdraw problem, such as sampling [13], dividing the map into blocks according to artificial rules [21] and proposing new chart types [33].

Dividing the map into blocks is to divide the map area according to certain rules, make statistics on the relevant data in each divided area, and use different methods to visualize the statistical results, such as Hexagonal Binning [39], choropleth map [11], bubble map [29], catogram [27], etc. These methods are based on the artificial rules, the adjacent points are aggregated into a single graphic element, which effectively reduces the visual clutter and covers the corresponding area of the original geospatial data. In order to visualize finer granularity, appropriate interpolation algorithm is used to display the data as a continuous visualization form, such as heatmap [14]. These graphs are easy to distinguish visually, while able to contain massive datasets and perform well dealing with overdraw problems. They not only retain the characteristics of data distribution but also has a good visual effect.

The sampling method is another strategy to enhance visual perception [18]. Different sampling schemes have the effect of simplifying scatter plot to a different extent, which is convenient for users to identify meaningful information from visualization. Ellis and Dix [16, 17] reduce the size of points by random sampling. Bertini et al. [7] proposed a non-uniform sampling method to preserve the relative density of scattered points. Z.Zhou et al. [58] proposes an attribute-based sampling method, which not only maintains the point densities, but also maintains the spatial autocorrelations.

To avoid misleading users by poor visualization effects, GeoVis introduces a method based on latent space to automatically select the most appropriate chart type by assuming that the main information of geospatial data is the dispersion of data.

3 Overview

In this section, we outline the challenges encountered in our research and the design goals, and summarize the pipeline of our visualization system.

3.1 Task Challenges

We are committed to building a visualization recommendation system to support the chart recommendation of geospatial data. Through literature research and analysis of the potential target audience of our visualization system, we finally summarized our relevant tasks as follows:

T1 **How to recommend a better chart type for a given data to reveal the distribution characteristics of the data?** Obviously, different visual charts play different roles in people's perception of data. An excellent visual work presentation can help users find the laws behind geospatial data more quickly. A typical example is that John Snow, a British doctor, found the cause of the cholera epidemic in time by mapping the Ghost Map of cholera cases. At present, the widely used mobile devices and sensors produce a large number of geospatial data. How to take the most appropriate visual charts to mine the laws has brought us new opportunities and challenges.

T2 **How to choose the appropriate map style and color according to different data?** The selection of map style and color matching is an important part of visual design. The expressive ability of visual charts depends on the designer's accurate application of map style and color matching. Poor style and color matching may reduce the audience's ability to recognize data. Therefore, how to choose the appropriate map style and color matching according to different data is an arduous task.

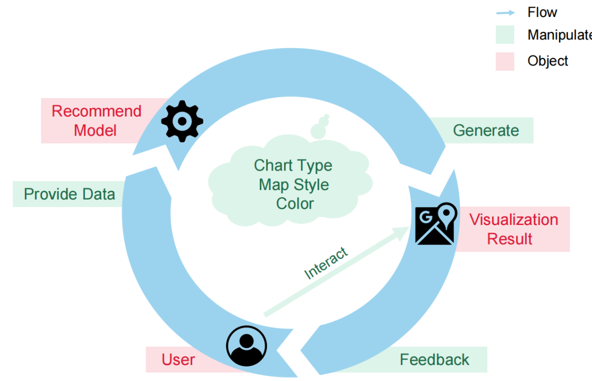

Fig. 2. Sample what our system does. Users can easily visualize the submitted data and edit the recommended results.

T3 **How to improve the use efficiency as much as possible?** With the continuous breakthrough of visualization technology, how to realize the intelligence and simplicity of visual recommendation system is an urgent problem to be solved. When a new dataset is given, if the visualization system can automatically display the characteristics of the data and give the corresponding visualization chart (example in Fig. 2), it will save the time cost of data analysts to a great extent and help to improve the use efficiency of users.

3.2 Design Considerations

Combining our goals with geographic visualization practice, we have established three main design goals to guide our system:

Data-Driven Back-end Model. An excellent visual chart can help users observe the characteristics of data more intuitively. How to recommend visual chart types for different input data and different tasks is the key link in data visualization. Up to now, no theory can well prove which visual chart type can better reveal the distribution characteristics of geographic data. At the same time, the rule-based visual recommendation method may be difficult to capture the nonlinear relationship between data through simple rules. Therefore, we hope that our chart recommendation method is to learn directly from the data and can be trained directly through the deep learning model. We also hope that our back-end model can also recommend map style and color based on a data-driven approach, which can improve the efficiency of dedicated user analysis in a wider range of scenarios.

Automation. Compared with other decision-making methods, an automatic method is more efficient and operable, and data perception and visual decision-making can be regarded as a whole system for target optimization. In order to improve the efficiency of the system and reduce the amount of user operations, people can get the most information through simple operations. We hope to build an automatic visualization recommendation system.

Support Multiple Interactions. The pattern of data can be vividly displayed through charts, which can form an intuitive feeling. The most important point is that different visual coding elements such as color or size can bring different visual differences, which can affect the characteristics of users' observation data. In order to achieve a very friendly user experience, we hope to provide a variety of interaction methods, taking into account the different needs of users, so that our system has flexibility. Users can slightly adjust the chart results recommended by the algorithm through simple operation, so as to mine more hidden information.

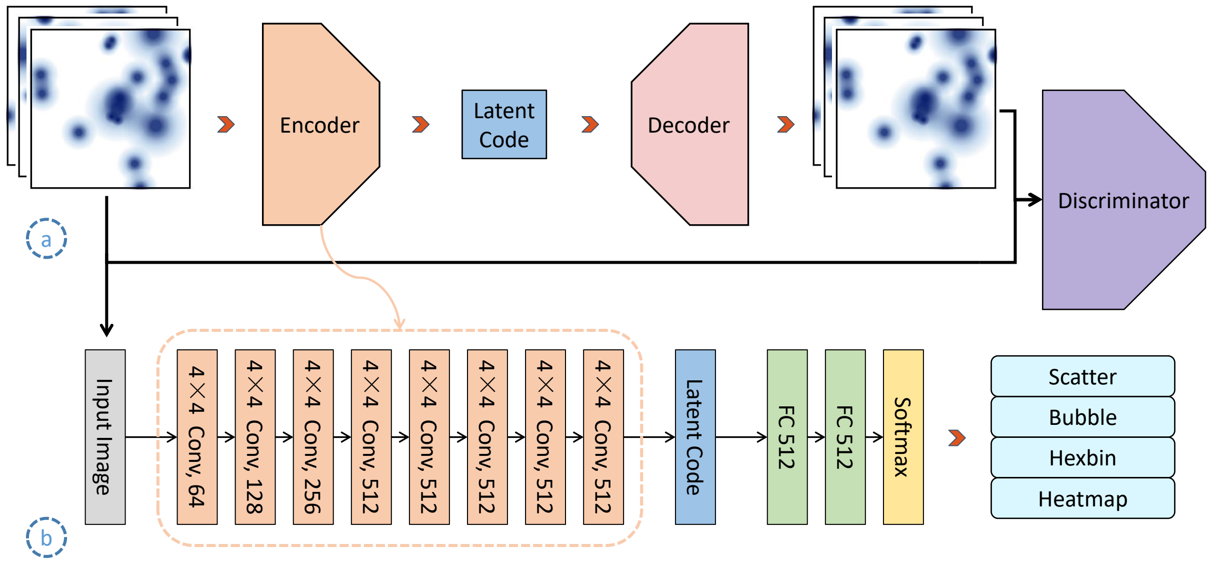

Fig. 3. This figure illustrates the flow of our model architecture.(a) is a schematic representation of our pre-trained encoder model. (b) is our implementation of the deep learning model recommended by map type. The map type candidates suggested from the latent code, which is calculated by the encoder. The input size required by our encoder is 256 $\times$ 256. In the convolutional layers, we use the 2D Convolutional layer, Batchnorm2d and LeakyRelu.

3.3 Visual Recommendation System

Due to the characteristics of geospatial data, there are many challenges to achieve the above tasks and design objectives in this work. First, Because there are not enough dataset-visualization pairs., we crawl the real geospatial data from different visualization communities, and crawl the corresponding visualization chart types as labels. To achieve augment data and improve the final effect of the model, we manually simulated some distribution laws and invited visualization experts to mark these datasets into different chart types. The purpose of expert evaluation is: whether the geospatial information characteristics of the data are accurately reflected, whether the focus of the information is expressed with an appropriate amount of data ink ratio, and whether the overall visualization results are beautiful. Finally formed 12000 data visualization type sample pairs. Second, under the assumption of complete spatial randomness (CSR), the original scatter diagram is only a simple expression of the distribution characteristics. In order to mine the deep spatial distribution characteristics and take into account the uncertainty of geospatial data, kernel density estimation is undoubtedly an effective method. Finally, considering various visual coding factors such as chart type, map style and map color, our method needs to get the optimal solution in a variety of permutations and combinations.

In this work, we propose an automatic visualization recommendation system to solve these problems. Fig.1 shows the pipeline of the GeoVis system. In the first stage, we use the quad-tree algorithm to optimize the bandwidth selection problem in kernel density estimation to ensure the accuracy of our estimation results. In the second stage, we encode the density map generated based on kernel density estimation with encoder, and reduce the dimension of the coding result. At the same time, we calculate the corresponding map style according to the aesthetics measures. In the third stage, we classify according to the coding results after dimensionality reduction and transform the recommendation problem into a ranking problem by classification. The classification result with the highest probability is our optimal visual chart type. In addition, we calculate the color of map visual elements based on the beauty of color. Finally, we improve our visualization system according to the guidance of experts, so that users can change the visualization effect through interaction.

4 Methods

4.1 Geospatial Feature Map Generation

When a large amount of data needs to be identified on the map, there will be a lot of overlap between points. Therefore, it is necessary to recommend different charts to show the overall distribution of geographic data. To make an accurate visual recommendation, we need to perceive the geographical distribution characteristics of spatial scattered points and abstract them into geospatial feature maps for expression.

Kernel density estimation (KDE) algorithm was originally proposed by Parzen [38]. It can clearly show the density of scattered points in different regions by fitting data points, which is a common spatial analysis technology in the field of Geographic Information System [57]. As a non-parametric density estimation method, KDE can randomly select samples from data to estimate density, which has strong practicability and flexibility. At present, KDE is also widely used in epidemiology, ecology and other fields.

Due to the strong dispersion and randomness of geographic information data, compared with other methods, the kernel density estimation method transforms the scattered points in geographic space into a smooth density map, which can better describe the distribution law of geospatial data, so as to make the calculation results more accurate. In this paper, we use data pair $\{x_i, y_i\}$ to represent a scattered point on a geographic map. The density of the original scatter data can be expressed as:

$$f_h\left(\hat{x}, y\right) = \frac{1}{nh_xh_y}\sum_{i=1}^{n} K\left(\left|\frac{x-x_i}{h_x}\right|\right)\left(\left|\frac{y-y_i}{h_y}\right|\right) \tag{1}$$

where n represents the total number of data points, $K(x)$ is the kernel function, h_x and h_y are the bandwidth values of the kernel function on the X and Y axes, i represents the ith sample. **Select of kernel function.** The Gaussian function is the most commonly used kernel function, which can generate smooth and continuously differentiable density function. Therefore, the Gaussian function is selected as the kernel function in this paper.

Bandwidth selection. Different bandwidth determines the smoothness of the function, which may produce completely different estimation results. Compared with kernel function, bandwidth has a more significant impact on the accuracy of kernel density estimation [45]. Therefore, to estimate the kernel density more accurately, the choice of

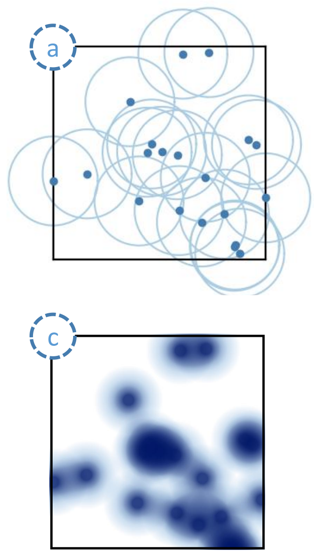

Fig. 4. This figure compares the results of different kernel density estimation methods. (a) Result of fixd KDE(rule of thumb, bandwidth = 0.607). (b) Result of adaptive KDE(ours). (c) The fixed bandwidth used to estimate density in (a). (d) The adaptive bandwidths used to estimate density in (b).

bandwidth is very important. Although too large bandwidth will make the density change tend to be smooth, it may ignore the small changes in local areas. A small bandwidth may add some unnecessary details. A suitable bandwidth should find a relative balance between the scattered dense area and the scattered sparse area. Due to the uneven distribution of geospatial data, the estimation result of bandwidth calculated by the rule of thumb [47] may not be accurate in practical application.

Adaptive kernel density estimation algorithm. As a classical spatial segmentation and indexing technology, quad-tree can segment the spatial region into four blocks with different densities by recursion. In this paper, the existing kernel density estimation methods are fully combined with the characteristics of geospatial data, and an adaptive bandwidth kernel density estimation algorithm based on quad-tree is proposed (Algorithm 1). The algorithm first divides the data area into four blocks, and then records the number of midpoint and side lengths of the segmented area. When the judgment conditions are executed for all regions, we can get the segmentation results and adaptively calculate the bandwidth. We use quad-tree segmentation to obtain different local density distribution patterns in the whole spatial region, so that the larger the data density, the more blocks are segmented, and the smaller the size of the blocks is.

The number and the side length of points in the data block are important conditions for our judgment. As we know, when n is too large, the algorithm can not well show the details of data distribution. When n is too small, the level of quad-tree segmentation will continue to deepen, resulting in a corresponding increase in the amount of calculation. Similarly, the setting of side length L is also to avoid the quad-tree from being divided into too deep levels and increase unnecessary calculation. After referring to the division theory in Knn-Kde [10], we set N_{Max} to $\sqrt{N}$, where N represents the total number of sample points, and set L_{min} to $\frac{L}{30}$, where L is the side length of the picture.

Execute the above quad-tree segmentation process until all regions do not meet the conditions of region segmentation. Through the number and size of blocks, it reflects the density of geospatial data to a certain extent. At the same time, we have sufficient reasons to believe that the use of smaller bandwidth in data-intensive places can more fully reflect the details of the region. Finally, the algorithm divides the spatial area of the original geographic data into data blocks of different sizes, and then obtains the local optimal bandwidth to achieve the purpose of adaptive bandwidth kernel density estimation. Further, we can use the generated variable bandwidth to draw the density map. Fig. 4 shows the different results of our method and thumb-rule under the same data.

4.2 Candidate Map Types Recommendation

To solve our visual recommendation task quickly and efficiently, we propose a deep learning model to solve our recommendation problem. This model takes the density map as input and realizes the mapping from density map to latent code by constructing a pre-trained encoder. Then it uses a classifier to convert the recommendation problem into a classification problem. The purpose is to obtain the calculated classification probability value as the basis for recommending geographic visualization types.

Encoder. The density map $x \in \mathbb{X}$ is different from other natural images. There is only the difference in brightness in the density map. As shown in the Fig. 3, in order to obtain high-quality latent code $z \in \mathbb{Z}$, we design and pre-train our encoder based on GANs [20]. In this model architecture, we can calculate the loss between the input of the encoder $E : \mathbb{X} \mapsto \mathbb{Z}$ and the output of the decoder $G : \mathbb{Z} \mapsto \mathbb{X}$, and the introduction of a discriminator $D : \mathbb{X} \mapsto \mathbb{R}_{[0,1]}$ has more strict requirements for the output of the decoder. At present, the model architecture based on GANs has also made great research progress in the field of image and vision.

There are two main goals of the encoder E. One is to generate the latent code of the density map, and the other is to reduce the dimension of the latent code $dim(z)$ as much as possible. We believe that to better classify geographic information data, we need to characterize the characteristics of the data. In order to mine the hidden deep-seated relationships under these features, we need to use the encoder to decouple these relationships, and use later code to represent the hidden features through a series of affine transformations. The main advantage of using latent code is that we can not only obtain the feature representation of geographic information data with high quality, but also have a strong learning ability for the deep-seated feature relationship, which is greatly conducive to obtaining better recommendation results. In order to achieve the above goal, we need to constrain the encoder to produce implicit variables that obey the Gaussian distribution.

In our encoder, the input size is 256 × 256. In the convolutional layers, we use the 2D convolutional layer, following by Batchnorm2d and LeakyRelu. The corresponding structure is also applied in our decoder. At the same time, the discriminator takes the original density map and the generated density map as input. For the stability of training, the discriminator only uses one convolutional layer as the last layer. Through the adversarial method, we can compare the high-quality density map generated by the decoder with the original density map, and use the discriminator to train our model. Compared with directly using the traditional encoder-decoder structure, this method can improve the representation quality of latent code for geographic information data and help to classify more accurately. We initially used the least squares loss function as the loss function of the discriminator, but found that the convergence of the model is relatively slow and the generated results have more noise. Therefore, we add a new loss in the form of L_1 distance. The L_1 distance is defined as follows:

$$L_{rec}(G) = \|x - G(x)\|_1 \tag{2}$$

$$L_{adv}(G) = logD(x) + log(1 - D(G(z))) \tag{3}$$

where latent code is sampled from standard gaussian distribution $z \sim N(0,1)$. Finally, the loss function combined with reconstruction loss and adversarial loss:

$$L_{generator}(G) = L_{adv}(G) + \lambda L_{rec}(G), \forall x \in X_{train} \tag{4}$$

Among them, λ is a hyperparameter, which is finally set to 200. Through many experiments, we find that the compression quality is higher when the latent code size is 512 dimensions. Therefore, we use the latent code with the size of 512 as the input of the classifier and train the weight of the encoder.

Classifier. Since it is a four classification task and the chart types to be classified are mutually exclusive, we consider using a softmax classifier. Softmax classifier is a typical algorithm applied to multi-classification problems. It has the advantages of easy training and high accuracy. It occupies an important position in the field of classification. For the latent code obtained by the encoder, the nonlinear expression ability of the model is further improved by three full connection layers. Finally, the results are output by the softmax classifier. In this paper, we use the cross-entropy function as our loss function, which is as follows:

$$L_{classification} = -\sum_{i=1}^{c} t_i log\left(y_i\right) \tag{5}$$

Where c is the number of classifications, y_i is the output of the ith sample, and t_i is the label of the ith sample. The cross-entropy function encourages correct classification and then optimizes each weight matrix by the back-propagation method. Finally, we calculate the probability of the four types according to the results of the classifier, and the ranking result is the category ranking recommended by our algorithm. The calculation method of probability is as follows:

$$p_i = \frac{e^{\hat{y}_i}}{\sum_k e^{\hat{y}_k}} (k = 0,1,2,3) \tag{6}$$

The result of sorting is the category ranking recommended by our algorithm. Through the above methods, we obtain the recommended candidates.

4.3 Candidate Design Choices Recommendation

For the selection of map style and color, we designed a serial calculation method to search our design space and help the system recommend the most appropriate geographic visualization display effect.

Map style selection. For geographic information visualization, the style of the map can show the overall color matching of the map and enrich the visual style. At present, there is no clear standard to guide the relationship between data distribution and map style. To better integrate data distribution features and map style, we refer to the work of Ngo et al [35], and define the following interface aesthetic features:

- **Balance degree.** It refers to the stability of the overall distribution of the original scattered points, and judges whether the upper and lower elements or the left and right elements are unbalanced:

$$feature_b = 1 - \frac{1}{2}\left(\frac{|S_L - S_R|}{max(S_L, S_R)} + \frac{|S_T - S_B|}{max(S_T, S_B)}\right) \tag{7}$$

  where $S_j (j = L, R, T, B)$ represents the area of interface space, and L, R, T, and B strand for left, right, top and bottom respectively.

- **Simplicity ratio.** It refers to the degree of directness and simplicity of the form of the density map:

$$feature_s = \frac{n_{block}}{n} \tag{8}$$

  Since the density map is a single channel, We calculate the number of blocks (n_{block}) in the density map using breadth-first traversals. n is the number of pixels on the density map.

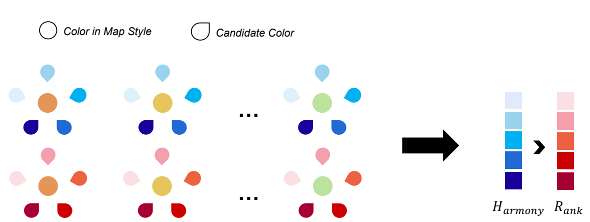

Fig. 5. This image shows the results of our calculations between street style and marquee colors using the color beauty formula. We obtained the color data for street style from the official map box website. Taking the comparison between red and blue as an example, we can see that blue is more suitable to be displayed in street style than red.

- **Density.** Density is the extent to which the map is covered with points:

$$feature_d = 1 - 2\left|0.5 - \frac{\sum_i^n a_i}{a_{frame}}\right| \tag{9}$$

  where a_i and a_{frame} are the sum of pixel i and density map, and n is the number of pixels on the density map.

To explore the hidden relationship between aesthetic factors and map style, we apply decision tree algorithm to real datasets to comprehensively consider each feature. We use scikit-learn for training. The decision tree is used as a classifier to fit features and labels. Max is set to 3 and criterion is set to Gini coefficient. Finally, we save the trained decision tree model for processing new data.

Color selection. As an important element of geographical visualization, color not only affects the readability of charts but also reflects the artistry and aesthetic value of charts. The rational use of color is conducive to improving the visual effect of geographical visualization charts, improving the transmission ability of geospatial information and reducing the cognitive burden of users. However, the color design in geographical visualization chart is highly professional, which requires professionals to have rich experience and profound insights into the cartographic theory and geospatial data. At the same time, due to the lack of unified color setting standards and the diversity of color space, the color recommendation of geographic visualization becomes more difficult and complex.

After identifying the map style , users need choose a color to apply. This paper introduces the calculation method of total variation($\mathscr{T}\mathscr{V}$) [2] into the generated color calculation, which is defined as the sum of the three channel differences between the colors in the color gamut and the colors that make up the map style. The color distance matrix **C** between candidate and backgrpund color is defined as follows:

$$\mathbf{C} = \{\delta_{i,j}\} = \delta(c_{cad}^{i}, c_{bac}^{j})^2 \tag{10}$$

where $\delta(c_{cad}^{i}, c_{bac}^{j})^2 = d_{color}(c_{cad}^{i} - c_{bac}^{j})$. This formula represents the difference between the candidate color and the background color of map style on different primary color channels. Normally, d_{color} is euclidean distance in rgb color space.

$\mathscr{T}\mathscr{V}$ can effectively search RGB color models and filter out colors with high contrast with map style. This calculation method will cover all color spaces. In order to avoid counterintuitive colors that are too bright or too dark, we define constraint:

$$\begin{cases} \overline{\varphi_{r,g,b}} > \alpha \\ \underline{\varphi_{r,g,b}} < \alpha \times \overline{\varphi_{r,g,b}} \end{cases} \tag{11}$$

where $\overline{\varphi_{r,g,b}}$ is calculated by the maximum value of three color channels divided 256, $\underline{\varphi_{r,g,b}}$ is calculated by the minimum value of three color channels divided 256, $\alpha = 0.5$. Using conditional constraints to restrict $\mathscr{T}\mathscr{V}$ can make the filtered color more in line with the perceptual effect of human eyes. By sorting the screening results of the above methods, we take the top five colors with the highest difference as the candidate colors. For generated candidate colors, this paper adopts Munsell theory [8] and uses hue, chroma, and value to represent color. By using Spencer's color aesthetic measure [34], to quantify the harmony between colors, and finally give the color ranking suitable for different styles.

The color aesthetic measure is:

$$M = O/C \tag{12}$$

where M is value of color aesthetic measure, O represents the number of elements of order and C is the number of elements of complexity.

Specifically, C is determined by the number of colors involved in quantization and the three attributes of color (hue, chroma and value):

$$C = C_m + C_h + C_v + C_c \tag{13}$$

Similarly, the calculation of O is also determined by the order factor between hue difference, value difference and chromaticity difference:

$$O = \sum O_h + \sum O_v + \sum O_c \tag{14}$$

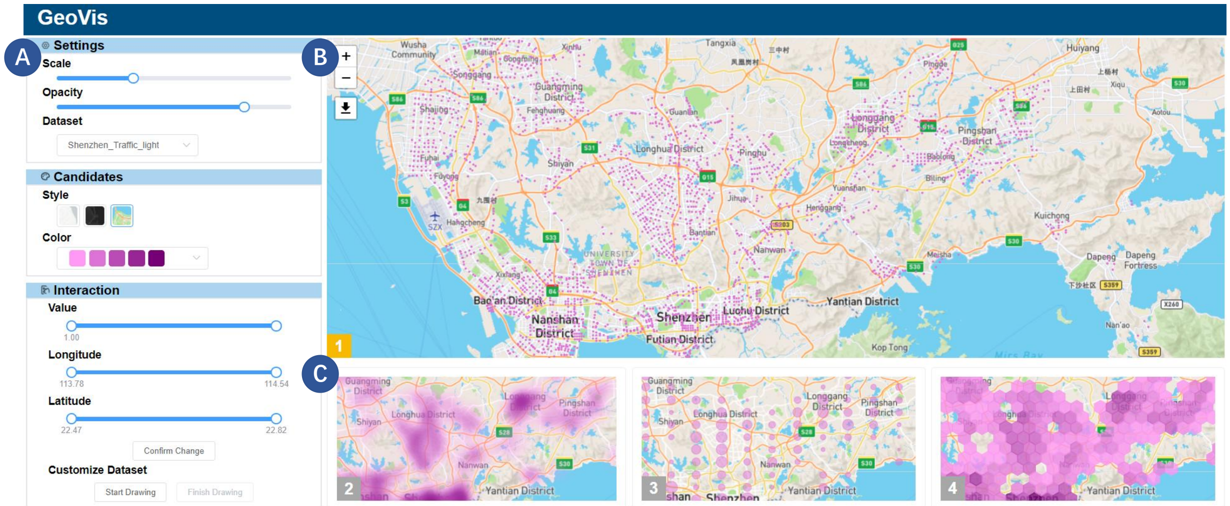

Fig. 6. This figure shows the overall interface of the recommendation system. Among them, part A shows the interactive panel of the system. Users can adjust the visualization effect by dragging and clicking. Part B shows GeoVis's most recommended visualization results. Part C provides other candidate chart types. Users can simply browse the selected visualization type in part B by clicking map type in C.

5 GeoVis System

GeoVis consists of two main modules, one is a web-based front-end user interface and the other is a back-end recommendation engine. Users who interact through the front-end interface can easily modify the recommended visual results. The back-end recommendation engine calculates the recommendation result of our algorithm according to the dataset returned by the front-end.

Fig. 6 shows the overall interface of the recommendation system, which is mainly divided into three windows: A, B and C. On the left is the interactive panel, which users can interact with when browsing the visual results. Users can adjust the visual scale and opacity by dragging the slider. Similarly, users can also use the filter module to screening the data. Dataset module provides users with different data options to be selected from. Users can easily view the recommendation results of different datasets. When users are not satisfied with the design choice recommended by the system, they can also click other options to change the visualization effect. Part B is the most recommended visualization result of the system, and the legend is automatically displayed in the lower right corner of the picture. At the same time, users can zoom and download the visual effect map. Part C provides other candidate recommended chart types. Users can switch the types by clicking the candidate chart with the mouse.

6 Evaluation

To prove the efficiency of our system GeoVis and make sure it has real-life usage scenarios,, this section presents the evaluation of the recommended method in terms of qualitative and quantitative evaluations.

6.1 Model Performance

GeoVis proposed by us is used to help users realize the visual recommendation of geographic information data, to support users' faster front-end interaction and data analysis. To evaluate our recommendation algorithm, we conducted some experiments to evaluate our recommendation results.

After the usefulness analysis of the latent code, we tried to quantitatively evaluate the effectiveness of our method. Considering the existing visual recommendation methods do not recommend geographic information data, we used the classical image classification model to conduct ablation experiments. We chose these algorithms because our recommendation task is realized through the classification model based on latent code. We want to observe whether other classification algorithms can achieve the same satisfactory performance. At the same time, in order to ensure that our adaptive KDE method is effective, we also compared the results of using the thumb rule with that of using the adaptive KDE. We hope to figure out whether the adaptive KDE method can perform better effect on the generation of latent code.

We built our recommendation model using PyTorch and tested the dataset mentioned in section 3.3 on a server using NVIDIA 3090. We did statistics and recorded the evaluation results (Table 1). It can be seen from Table 1 that the KDE method proposed in section 4.1 is effective. Results got from different bandwidth settings is better than that of fixed bandwidth in almost all models. It shows that the adaptive kernel density estimation method is helpful to better mine the distribution characteristics of geographic information data and improve the accuracy. We note that our recommendation model based on latent code is better than other classification models in accuracy. This result is within our expection. Because latent code can learn the deep-seated distribution feature relationship, thus conclusively speaking our model can have a more effective feature representation for geographic information data.

6.2 Case Study

In order to further prove the reliability of this recommendation algorithm, we used the real dataset in the travel website to verify our recommendation results. The dataset collected the GPS information data of 8375 travelers. We filtered the abnormal information outside the longitude and latitude range and the data of non walking marks, and finally obtained 6247 pieces of distribution data with longitude and latitude.

To compare whether domain experts prefer our model for the same task, we propose a query task. In the Spatial Task defined by TaskVis [46], We have assigned the task of querying the street congestion in Hangzhou, in order to help the decision-makers in the traffic field timely dispatch resources. We also invited domain experts to make suggestions on the recommendation results of this task.

Inspired by TaskVis, we treat geographic information data as a combination of multiple numerical data attribute columns, and then take Reverse-complexity-based ranking in TaskVis to obtain the most recommended map types.

Fig. 7 shows the distribution of tourists in Hangzhou. Among them, GeoVis takes the hexbin type as the result of rank 1, which is in line with the experts' prior recommendations, and the heatmap, bubble map and scatter are the second, third and fourth candidates in turn. At the same time, according to the system's map style and color mark selection algorithm, the light style and green based color palette are finally presented. In contrast, the rule-based approach recommends bubble charts.

Table 1. This table shows the predicted results of adaptive KDE versus fixed bandwidth KDE. At the same time, we compare different deep learning models with our results based on latent code in recommendation tasks.

MODEL	Fixed Bandwidth KDE								Adaptive KDE							
	10	100	500	1000	5000	10000	50000	100000	10	100	500	1000	5000	10000	50000	100000
Random	30.4%	22.6%	26.2%	23.6%	24.8%	21.4%	23.3%	19.2%	26.2%	28.4%	25.2%	26.4%	25.5%	24.6%	25.7%	25.2%
VGG16	26.9%	54.5%	64.9%	63.2%	60.5%	59.1%	56.0%	52.9%	27.1%	55.4%	68.2%	65.1%	64.6%	62.2%	57.5%	54.8%
GoogLeNet	26.8%	58.7%	69.2%	64.1%	62.6%	61.2%	58.7%	55.8%	27.9%	59.0%	72.1%	70.5%	69.2%	68.3%	65.8%	60.5%
ResNet	24.1%	58.9%	74.3%	72.9%	72.7%	70.8%	64.2%	59.6%	28.2%	59.5%	76.4%	72.8%	71.9%	70.5%	66.3%	60.6%
DenseNet	28.8%	61.1%	78.5%	77.1%	75.3%	74.6%	68.1%	62.5%	29.8%	63.2%	82.5%	80.9%	76.6%	75.9%	70.1%	68.5%
Ours	**30.3%**	**69.5%**	**84.6%**	**84.3%**	**82.5%**	**80.1%**	**76.7%**	**74.0%**	**31.3%**	**73.9%**	**92.8%**	**92.3%**	**91.9%**	**90.6%**	**87.6%**	**85.4%**

(a) Rank1: hexbin

(b) Rank2: heatmap

(c) Rank3: bubble(result of rule-based method)

(d) Rank4: scatter

Fig. 7. Illustration of GeoVis recommendation results. Hexbin is the map type recommended most by GeoVis. In general, through GeoVis, we can easily observe that Hangzhou tourists are mainly concentrated around the West Lake of Hangzhou, and the Yuewang temple and the dam near the West Lake are the preferred areas for tourists.

By observing the visualization, the experts concluded that both recommendation results can reflect the congestion distribution characteristics of Hangzhou to some extent. As for the recommendation results, experts think that for the dataset with a large number of data points, even if the transparency of the scatter is adjusted, it is still difficult to identify, and user can't judge the distribution characteristics of the region visually. The bubble map shows the density of local areas through the bubble radius. Compared with the scatter map, the bubble map can more succinctly show the distribution of dataset, but there will still be overlap. Moreover, the aggregated bubble center is not strictly presented according to the original distribution, which affects the user's perception results. It is worth noting that both the heatmap and the hexbin better depict the overall distribution characteristics and intuitively present the densely populated areas. Compared with the visualization results of the heatmap, experts believe that the hexbin can more accurately reflect the distribution information of the data, and at the same time, it also describes the continuity trend of the data distribution. Therefore, hexbin can reduce the decision time of managers more than bubble map, which helps managers to deploy resources.

Overall, the experts were more receptive to our data-driven approach through the qualitative case study. In addition, we found in our tests that since rule-based recommendation systems use a simpler formula, this can lead to consistent scores and no ranking. In contrast, our approach avoids this problem by using a latent code approach that is well suited to complex nonlinear relationships and thus gives more meaningful ranking results.

7 Conclusion

This paper focuses on the graph recommendation of geographic visualization. We assume that the major feature of geographic information data is the distribution characteristics of data, and design an automatic visual recommendation system. Specifically, an adaptive kernel density estimation algorithm based on quad-tree is proposed to obtain the density map of geographic information data. Then we get the latent code of density map through the pre-trained encoder. For latent code, we use a classifier to get the final recommendation result. At the same time, we use the aesthetic measure to calculate the style and color attributes of the map. Finally, we show our recommendation results in the front-end interface. Through design evaluation experiments, we verify the feasibility and effectiveness of our recommended algorithm in different application scenarios.

Our work has taken a small step in exploring the recommendation system related to geographic visualization. However, we believe that such problems still have a long way to go. In future work, we plan to study how to make personalized recommendations according to different user preferences. In addition, we will consider more visual encoding to facilitate more efficient visualization and data analysis for users and how to make recommendations for relational geographic data.

References

[1] A. Acedo, M. Painho, S. Casteleyn, and S. Roche. Place and city: toward urban intelligence. *ISPRS International Journal of Geo-Information*, 7(9):346, 2018.

[2] H. A. Aly and E. Dubois. Image up-sampling using total-variation regularization with a new observation model. *IEEE Transactions on Image Processing*, 14(10):1647–1659, 2005.

[3] G. Andrienko, N. Andrienko, C. Boldrini, G. Caldarelli, P. Cintia, S. Cresci, A. Facchini, F. Giannotti, A. Gionis, R. Guidotti, et al. (so) big data and the transformation of the city. *International Journal of Data Science and Analytics*, pp. 1–30, 2020.

[4] G. Andrienko, N. Andrienko, C. Boldrini, G. Caldarelli, P. Cintia, S. Cresci, A. Facchini, F. Giannotti, A. Gionis, R. Guidotti, et al. (so) big data and the transformation of the city. *International Journal of Data Science and Analytics*, pp. 1–30, 2020.

[5] G. Andrienko, N. Andrienko, S. Drucker, J.-D. Fekete, D. Fisher, S. Idreos, T. Kraska, G. Li, K.-L. Ma, J. Mackinlay, et al. Big data visualization and analytics: Future research challenges and emerging applications. In *BigVis 2020: Big Data Visual Exploration and Analytics*, 2020.

[6] N. Andrienko, G. Andrienko, G. Fuchs, A. Slingsby, C. Turkay, and S. Wrobel. *Visual analytics for data scientists*. Springer International Publishing, 2020.

[7] E. Bertini and G. Santucci. By chance is not enough: preserving relative density through nonuniform sampling. In *Proceedings. Eighth International Conference on Information Visualisation, 2004. IV 2004.*, pp. 622–629. IEEE, 2004.

[8] F. Birren. A grammar of color, a basic treatise on the color system of albert h. munsell. 1969.

[9] M. A. Borkin, A. A. Vo, Z. Bylinskii, P. Isola, S. Sunkavalli, A. Oliva, and H. Pfister. What makes a visualization memorable? *IEEE transactions on visualization and computer graphics*, 19(12):2306–2315, 2013.

[10] L. Breiman, W. Meisel, and E. Purcell. Variable kernel estimates of multivariate densities. *Technometrics*, 19(2):135–144, 1977.

[11] C. A. Brewer and L. Pickle. Evaluation of methods for classifying epidemiological data on choropleth maps in series. *Annals of the Association of American Geographers*, 92(4):662–681, 2002.

[12] W. Chen, F. Guo, and F.-Y. Wang. A survey of traffic data visualization. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):2970–2984, 2015.

[13] X. Chen, T. Ge, J. Zhang, B. Chen, C.-W. Fu, O. Deussen, and Y. Wang. A recursive subdivision technique for sampling multi-class scatterplots. *IEEE transactions on visualization and computer graphics*, 26(1):729–738, 2019.

[14] M. DeBoer. Understanding the heat map. *Cartographic perspectives*, (80):39–43, 2015.

[15] V. Dibia and Ç. Demiralp. Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks. *IEEE computer graphics and applications*, 39(5):33–46, 2019.

[16] A. Dix and G. Ellis. By chance enhancing interaction with large data sets through statistical sampling. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pp. 167–176, 2002.

[17] G. Ellis and A. Dix. Density control through random sampling: an architectural perspective. In *Proceedings Sixth International Conference on Information Visualisation*, pp. 82–90. IEEE, 2002.

[18] G. Ellis and A. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE transactions on visualization and computer graphics*, 13(6):1216–1223, 2007.

[19] M. Eremia, L. Toma, and M. Sanduleac. The smart city concept in the 21st century. *Procedia Engineering*, 181:12–19, 2017.

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[21] F. Heimerl, C.-C. Chang, A. Sarikaya, and M. Gleicher. Visual designs for binned aggregation of multi-class scatterplots. *arXiv preprint arXiv:1810.02445*, 2018.

[22] H. H. Howard, R. B. McMaster, T. A. Slocum, and F. C. Kessler. Thematic cartography and geovisualization. 2008.

[23] K. Hu, M. A. Bakker, S. Li, T. Kraska, and C. Hidalgo. Vizml: A machine learning approach to visualization recommendation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2019.

[24] K. Hu, D. Orghian, and C. Hidalgo. Dive: A mixed-initiative system supporting integrated data exploration workflows. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, pp. 1–7, 2018.

[25] Y. Hu, S. Gao, D. Lunga, W. Li, S. Newsam, and B. Bhaduri. Geoai at acm sigspatial: progress, challenges, and future directions. *Sigspatial Special*, 11(2):5–15, 2019.

[26] J. Jacobs and B. Rudis. *Data-driven security: analysis, visualization and dashboards*. John Wiley & Sons, 2014.

[27] D. A. Keim, C. Panse, and S. C. North. Medial-axis-based cartograms. *IEEE computer graphics and applications*, 25(3):60–68, 2005.

[28] H. Li, Y. Wang, S. Zhang, Y. Song, and H. Qu. Kg4vis: A knowledge graph-based approach for visualization recommendation. *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[29] M. Li, F. Choudhury, Z. Bao, H. Samet, and T. Sellis. Concavecubes: Supporting cluster-based geographical visualization in large data scale. In *Computer Graphics Forum*, vol. 37, pp. 217–228. Wiley Online Library, 2018.

[30] Y. Luo, X. Qin, N. Tang, and G. Li. Deepeye: Towards automatic data visualization. In *2018 IEEE 34th international conference on data engineering (ICDE)*, pp. 101–112. IEEE, 2018.

[31] R. Ma, H. Mei, H. Guan, W. Huang, F. Zhang, C. Xin, W. Dai, X. Wen, and W. Chen. Ladv: Deep learning assisted authoring of dashboard visualizations from images and sketches. *IEEE transactions on visualization and computer graphics*, 27(9):3717–3732, 2020.

[32] J. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. *IEEE transactions on visualization and computer graphics*, 13(6):1137–1144, 2007.

[33] A. Mayorga and M. Gleicher. Splatterplots: Overcoming overdraw in scatter plots. *IEEE transactions on visualization and computer graphics*, 19(9):1526–1538, 2013.

[34] P. Moon and D. E. Spencer. Aesthetic measure applied to color harmony. *JOSA*, 34(4):234–242, 1944.

[35] D. C. L. Ngo, L. S. Teo, and J. G. Byrne. Modelling interface aesthetics. *Information Sciences*, 152:25–46, 2003.

[36] M. Nöllenburg. Geographic visualization. In *Human-centered visualization environments*, pp. 257–294. Springer, 2007.

[37] P. K. M. Owonibi. A review on visualization recommendation strategies. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)*, 2017.

[38] E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[39] E. Polisciuc, C. Maçãs, F. Assunção, and P. Machado. Hexagonal gridded maps and information layers: a novel approach for the exploration and analysis of retail data. In *SIGGRAPH ASIA 2016 Symposium on Visualization*, pp. 1–8, 2016.

[40] X. Qian, R. A. Rossi, F. Du, S. Kim, E. Koh, S. Malik, T. Y. Lee, and J. Chan. Ml-based visualization recommendation: Learning to recommend visualizations from data. *arXiv preprint arXiv:2009.12316*, 2020.

[41] X. Qin, Y. Luo, N. Tang, and G. Li. Deepeye: An automatic big data visualization framework. *Big data mining and analytics*, 1(1):75–82, 2018.

[42] X. Qin, Y. Luo, N. Tang, and G. Li. Making data visualization more efficient and effective: a survey. *The VLDB Journal*, 29(1):93–117, 2020.

[43] B. Rankin. Cartography and the reality of boundaries. *Perspecta*, 42:42–45, 2010.

[44] S. Santala. Fast interactive design of scatterplots for large data set visualisation. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–6, 2020.

[45] D. W. Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[46] L. Shen, E. Shen, Z. Tai, Y. Song, and J. Wang. Taskvis: Task-oriented visualization recommendation. In *Proc. EuroVis*, vol. 21, 2021.

[47] B. W. Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

[48] L. A. Treinish. Task-specific visualization design. *IEEE Computer Graphics and Applications*, 19(5):72–77, 1999.

[49] J. J. Van Wijk. The value of visualization. In *VIS 05. IEEE Visualization, 2005.*, pp. 79–86. IEEE, 2005.

[50] M. Vartak, S. Huang, T. Siddiqui, S. Madden, and A. Parameswaran. Towards visualization recommendation systems. *ACM SIGMOD Record*, 45(4):34–39, 2017.

[51] Q. Wang, Z. Chen, Y. Wang, and H. Qu. Applying machine learning advances to data visualization: A survey on ml4vis. *arXiv preprint arXiv:2012.00467*, 2020.

[52] G. Wills and L. Wilkinson. Autovis: automatic visualization. *Information Visualization*, 9(1):47–69, 2010.

[53] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE transactions on visualization and computer graphics*, 22(1):649–658, 2015.

[54] A. Wu, Y. Wang, X. Shu, D. Moritz, W. Cui, H. Zhang, D. Zhang, and H. Qu. Survey on artificial intelligence approaches for visualization data. *arXiv preprint arXiv:2102.01330*, 2021.

[55] A. Wu, Y. Wang, M. Zhou, X. He, H. Zhang, H. Qu, and D. Zhang. Multivision: Designing analytical dashboards with deep learning based recommendation. *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[56] J. XIA, J. LI, and S. CHEN. A survey on interdisciplinary research of visualization and artificial intelligence. *Scientia Sinica (Informationis)*, 51:1777–1801, 2021.

[57] P. Zhao, X. Liu, J. Shen, and M. Chen. A network distance and graph-partitioning-based clustering method for improving the accuracy of urban hotspot detection. *Geocarto International*, 34(3):293–315, 2019.

[58] Z. Zhou, X. Zhang, Z. Yang, Y. Chen, Y. Liu, J. Wen, B. Chen, Y. Zhao, and W. Chen. Visual abstraction of geographical point data with spatial autocorrelations. In *2020 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 60–71. IEEE, 2020.

[59] T. Zou, W. Li, P. Liu, X. Su, H. Huang, Y. Han, and X. Guo. An overview of geospatial information visualization. In *2018 IEEE International Conference on Progress in Informatics and Computing (PIC)*, pp. 250–254. IEEE, 2018.