ScatterNet: A Deep Subjective Similarity Model for Visual Analysis of Scatterplots

Yuxin Ma, Anthony K. H. Tung, Wei Wang, Xiang Gao, Zhigeng Pan, Wei Chen

Abstract—Similarity measuring methods are widely adopted in a broad range of visualization applications. In this work, we address the challenge of representing human perception in the visual analysis of scatterplots by introducing a novel deep-learning-based approach, ScatterNet, captures perception-driven similarities of such plots. The approach exploits deep neural networks to extract semantic features of scatterplot images for similarity calculation. We create a large labeled dataset consisting of similar and dissimilar images of scatterplots to train the deep neural network. We conduct a set of evaluations including performance experiments and a user study to demonstrate the effectiveness and efficiency of our approach. The evaluations confirm that the learned features capture the human perception of scatterplot similarity effectively. We describe two scenarios to show how ScatterNet can be applied in visual analysis applications.

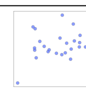
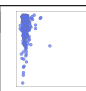
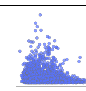
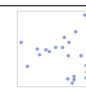
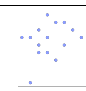
Index Terms—Scatterplot, similarity measuring, deep learning, visualization, visual exploration.

1 Introduction

SCATTERPLOTS [1] and scatterplot matrices (SPLOM) are widely used representations for depicting high-dimensional data in 2D or 3D space. When the dimension number increases, the usability of SPLOM decreases drastically [2], [3]. A variety of automated approaches [4], [5], [6] have been designed to retrieve informative views from huge amounts of plots to reduce the number of scatterplots being shown. The key for a successful retrieval is a well-defined similarity measure, which computes how similar two distinct views are by means of quantitative similarity values. An appropriate measuring method can not only allow for automatic retrieval of scatterplots, but also support visual querying, investigation and exploration of scatterplots with specific data distributions or potentially interesting patterns contained in the underlying dataset.

Existing solutions of similarity computing methods [6], [7] usually summarize a set of feature descriptors based on the input data or the rendered image of a scatterplot. For example, Scagnostics [7] defines nine hand-crafted feature descriptors to characterize scatterplots quantitatively from multiple interpretable perspectives such as data distributions, density, geometry, etc. The feature vectors of scatterplots computed with the descriptors above span a nine-dimensional feature space, and similarities of scatterplots can be defined based on metric distances among corresponding feature vectors. One of the issues in hand-crafted feature descriptors is that the descriptors and the derived similarity measure can sometimes fail to capture some patterns, especially for those related to human visual perception [8]. Table 1 illustrates an example in which the Scagnostics feature vectors of five scatterplots are shown. In the table, a query and four other plots (SC1 to SC4) that are ranked in ascending order of their distance to the query are shown. Contrary to the ranking, SC3 and SC4 are perceived to be closer to the query based on visual perception.

TABLE 1
An example of Scagnostics with feature values of five scatterplots. SC1, SC2, SC3 and SC4 are ranked in ascending order of the Euclidean distance to the query. Based on the distances, SC1 and SC2 are close to the query in terms of Scagnostics features, however SC3 and SC4 are more similar to the query scatterplot than SC1 and SC2 from the perspective of visual perception, which indicates that in this case the distances of Scagnostics features fail to reflect the visual similarity.

					
Features	**Query**	**SC1**	**SC2**	**SC3**	**SC4**
Outlying	0.382	0.471	0.236	0.000	0.441
Skewed	0.572	0.805	0.781	0.686	0.780
Clumpy	0.154	0.038	0.012	0.179	0.260
Sparse	0.195	0.036	0.022	0.243	0.336
Striated	0.100	0.034	0.044	0.048	0.083
Convex	0.249	0.321	0.343	0.106	0.022
Skinny	0.581	0.476	0.604	0.673	0.265
Stringy	0.266	0.266	0.301	0.422	0.244
Monotonic	0.004	0.001	0.016	0.049	0.051
Euclidean Distances to Query		**0.348**	**0.358**	**0.469**	**0.482**

In this paper, we investigate the design and usage of an effective approach for measuring similarities of scatterplots to support effective and efficient visual querying and exploration of scatterplots. Our approach is motivated by the successful applications of similarity measures and recent studies on applying knowledge of perception in visual analytics [9], [10], [11]. We believe that capturing human perception on similarities of scatterplots can be significantly important in many applications such as searching, exploration, and temporal trend analysis of large numbers of plots.

To this end, we propose a novel approach, ScatterNet, for modeling subjective similarity by utilizing human visual perception information. The core idea is to employ human-labeled judgment

- *Yuxin Ma and Wei Chen are with the State Key Lab of CAD&CG, Zhejiang University, China. Email: mayuxin@zju.edu.cn, chenwei@cad.zju.edu.cn. Wei Chen is the corresponding author.*
- *Anthony K. H. Tung and Wei Wang are with National University of Singapore. Email: {atung,wangwei}@comp.nus.edu.sg.*
- *Xiang Gao and Zhigeng Pan are with Hangzhou Normal University, China. Email: geekplux@gmail.com, zgpan@cad.zju.edu.cn.*

Manuscript received xx xx, 201x; revised xx xx, 201x.

on scatterplot similarities as training data, and utilize state-of-the-art deep neural networks to construct features from the plot images automatically. The Convolutional Neural Networks (CNNs) are able to learn rich semantic features from large scale data by tuning its internal parameters. The learned features are then used for computing similarities by utilizing the Euclidean distances. Consequently, ScatterNet is able to effectively characterize similarities between scatterplots by considering human perception information. To the best of our knowledge, this paper is the first one that leverages deep-learning-based image recognition methods to enhance the understanding and exploration of scatterplots.

In summary, our work presents two contributions:

- A novel approach for characterizing perception-based similarity between scatterplots quantitatively;
- A deep-learning-based method, ScatterNet, to generate a set of neural network layers in order to transform scatterplots into feature vectors for similarity computation.

The remaining sections are organized as follows. Related work is covered in Section 2. Section 3 introduces the model and its building process. We evaluate ScatterNet withcases and user studies in Section 4. Section 5 presents discussions and limitations, followed by conclusions in Section 6.

2 Related Work

Our work is related to two broad topics: 1) visual quality and similarity metrics, and 2) perception-based quality metrics.

2.1 Visual Quality and Similarity Metrics

Visual quality and similarity metrics of plots have been intensively studied for many years. Bertini et al. [2] performed a comprehensive study and presented a systematic taxonomy on existing visual quality metrics. Generally, they can be divided into two categories: 1) data-based approaches, and 2) image-based approaches.

Data-based Approaches compute feature vectors from the input data. Inspired by the Cognostics [12], [13] which utilizes computers to guide diagnostics of plots, Wilkinson et al. [7] proposed a set of feature descriptors named Scagnostics (Scatterplot Diagnostics) which presents the data distribution in a 2D plot based on graph theory. The two-dimensional Scagnostics can be extended to three-dimensional counterparts [14]. In addition to applying Scagnostics for plot retrieval of large-scale SPLOMs, the Scagnostics feature descriptors can be utilized in many other applications. For instance, ScagExplorer [15] supported visual exploration of patterns appearing in the plots based on Scagnostics features. TimeSeer [16] revealed hidden temporal patterns and dynamics of data distributions from scatterplot time-series. Dang et al. [17] addressed the issue of scaling-variant characteristics in Scagnostics and presented a method to overcome scaling transformation of specific patterns. By combining human visual feedback with data-based diagnostics, Behrisch et al. [18] introduced a "feedback-driven view space exploration framework" as a guide of querying and exploration in large-scale scatterplot datasets. Anand et al. [19] used some specific Scagnostics features such as *Skewed* and *Monotonic* patterns to partition multivariate datasets and detect interesting patterns within multiple views.

Besides Cognostics for scatterplots, many works focused on designing data-based quality and similarity metrics for effective visual understanding. The rank-by-feature framework in [20] was designed to provide an interactive visual interface for exploring multiple kinds of plots and ranking them with various feature detection criteria. Sips et al. [21] presented two quality measures to quantify class consistency with class center gravity and entropies of spatial distributions. DimScanner [22] addressed the challenge of high workload to analyze a multitude of statistical charts derived from high-dimensional datasets and contributed a data structuring scheme for modeling the relations and disclosing redundant information among different charts.

Image-based Approaches are designed to analyze patterns by regarding plots as images, benefitting from the progress of image processing approaches in computer vision [23], [24], [25], [26]. There were some studies that employed image-density-based methods [4], [6], [27] to recognize and rank desired linear or non-linear patterns in scatterplots. Shao et al. [5] proposed a motif-based matching and ranking scheme to facilitate querying of specific patterns with a set of basic image patches extracted from existing scatterplots.

For other visualization forms, Pargnostics [28] extended the concept of Cognostics to parallel coordinate plots (PCPs) by using pixel-space features to assess the visual quality of PCPs. Behrisch et al. [29] proposed a methodology of discovering feature descriptors for adjacency matrices and concluded a suite of operational descriptors such as blocks, local binary patterns and edges. As a generalized method, Pixnostics [30] analyzed pixels in plot images and estimated their values for specific visualization tasks.

In recent years, some works tended to perform classification analysis and visual feature augmentation by extracting image features with machine-learning-based techniques. Reda et al. [31] proposed a technique to automate the visual detection process in large amounts of views. For statistical charts, ReVision [32] utilized support vector machines to recognize chart types presented in images. Similar works were presented in [33] and [34] where higher classification accuracy was achieved with deep neural networks.

Our approach is inspired by the successful application of deep neural networks in [33] and [34]. The scatterplots are converted into images, upon which a similarity measure is constructed. Without hand-crafted image features, we utilize the power of deep neural networks to extract subjective features automatically from large amounts of labeled data. The learned features are more specific and precise than hand-crafted feature descriptors from the perspective of representing hard-to-quantify human perception.

2.2 Perception-based Quality Metrics

While data-based approaches have been widely applied, in recent years visual perception was brought into the limelight as an alternative direction to study quality and similarity metrics. Sedlmair et al. [11] summarized a taxonomy of visual cluster separation factors in scatterplots and evaluated two quantitative measures [6], [21] anchored in the proposed factors. The approach in [35] was intended to specify task-dependent quality metrics and performed interactive dimensionality reduction to reveal patterns from subjective perspectives. Rensink et al. [36] performed a user study to evaluate how visual perception is related to estimating the correlations within scatterplots. In the meantime, Harrison et al. [9] conducted a large-scale experiment to investigate whether Weber's Law can model the precision of visual perception in judging data correlation in nine different visualization forms. The model was later improved in [10] by using a modified version derived from Weber's Law. In a more recent work in [37], Rensink indicated that Fechner's Law also holds in judging correlations. Additionally, the experiment showed that observers are likely to "perceive the information entropy in an image".

The "label-and-model" strategy was shared among a variety of existing works on visualization and human-computer interaction in order to capture perceptual information from human-defined labels. Albuquerque et al. [38] used the labeling strategy to capture users' annotations on task relatedness of scatterplots. A similar user study was carried out in [8] to evaluate the difference between Scagnostics and perceptual similarity. A number of suggestions on designing perceptually-balanced quality and similarity measures were given. Similar to kernel functions in machine learning, Perceptual Kernels were introduced in [39] to estimate perceptual differences among visual variables based on multiple visual channels such as color, shape, size, and their combinations with crowd-sourcing experiments. In this paper, we leverage users' labels for modeling similarities among different scatterplots with deep metric learning models.

3 Model and Construction

Figure 1 presents the pipeline for model construction. It consists of three main stages:

1) **Sample Collection and Generation** First, multiple public datasets are collected to generate scatterplot images. We propose a generation and sampling strategy to create effective unlabeled sets of scatterplots.
2) **Scatterplot Image Triplets Labeling** In this stage, we ask annotators to select similar scatterplot images and dissimilar ones for a set of anchor images. The results are converted into a set of triplets with each triplet consisting of an anchor scatterplot image, a similar image and a dissimilar image. Additionally, we perform a preliminary label analysis to explore the human-annotated labels and gain insights for the model design.
3) **Model Building** With the labeled triplets, we build a deep neural network to model the similarity among scatterplot images. A set of convolutional neural network (CNN) layers is trained as a feature extraction module to transform scatterplot images into feature vectors.

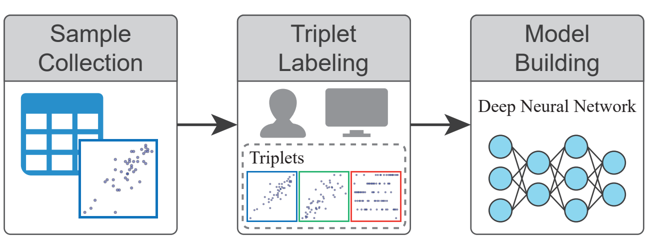


Fig. 1. The modeling pipeline consists of three main stages: 1) sample collection, 2) triplet labeling, and 3) model building.

3.1 Collecting and Generating Samples

Preparing effective samples requires 1) collecting datasets that maximize coverage over different types of scatterplots; 2) generating unlabeled sets for labeling.

3.1.1 Preparation of Scatterplot Images

Motivated by the data selection strategy in [8], we use the datasets from PyDataset library[1] that contains 757 datasets selected from Rdatasets[2]. The reason for not synthesizing datasets is that those synthetic patterns that rarely appear in real-world datasets may bias the results when human annotators identify the data distributions. For each dataset, we combine all possible pairs of columns to form a scatterplot, yielding $C_n^2 = \frac{n(n-1)}{2}$ scatterplots for a data table with n columns. Before plotting the scatterplots, we perform a data cleaning procedure to remove columns with invalid values and duplicated columns.

In plotting the column pairs, the canvas size is set to 200×200 pixels with a white background and an inner margin of 10 pixels. The dots are rendered in $RGB(0,0,255)$ with a radius of 2 pixels and opacity of 0.4. It should be noted that this setting is used only for the images that are labeled by human annotators. For the steps concerning building deep learning models, the scatterplots are re-plotted into gray-scale images. The point color is set to black, and the size and opacity are not changed.

We collect 50677 scatterplot images in total from the datasets. The corresponding Scagnostics features are computed and stored.

3.1.2 Generation of Unlabeled Sets

Before introducing the generation process, the terms used in the following sections are provided in Table 2:

TABLE 2
Definitions of terms used in describing our model building procedure.

Anchor Scatterplot/Image	The referred standard scatterplot
Candidate Scatterplot/Image	A set of images that are employed for comparisons with an anchor
Positive/Negative Image	The identified candidates that are similar/dissimilar to the anchor image
Triplet	A combination of an anchor image, a positive image, and a negative image
Unlabeled Set	The basic unit of the labeling task that consists of an anchor image and associated candidate images
Annotators	The ones who perform the labeling task
Query Scatterplot/Image	A selected scatterplot as a reference in the context of k-NN search

For each unlabeled set, the annotator is required to mark several most similar and dissimilar scatterplots as positive and negative examples. To limit labeling time and ensure annotators' concentration, 30 candidate scatterplots are provided for each unlabeled set. To ensure the effectiveness and efficiency of the labeling stage, we adopt the following principles:

1) Maximizing Diversity of Anchors: The diversity of anchor scatterplots should be as high as possible to cover a wide range of patterns;
2) Reducing Uncertainty: Some scatterplots may be too complex to be easily distinguished from others. These "hard examples" [40] can significantly improve the training effectiveness, efficiency, and stability of deep metric models to capture as much information as possible from the labels, i.e. reduce the prediction uncertainty of the model.
3) Improving Effectiveness of Candidates: For the anchor scatterplot in an unlabeled set, the possibility of containing similar and dissimilar scatterplots should be relatively high. This principle is intended to avoid the situation that in an unlabeled set all candidates are visually significantly different from or the same as the anchor scatterplot, making it hard to select positive or negative scatterplots for annotators.

1. https://github.com/iamaziz/PyDataset

2. https://vincentarelbundock.github.io/Rdatasets/

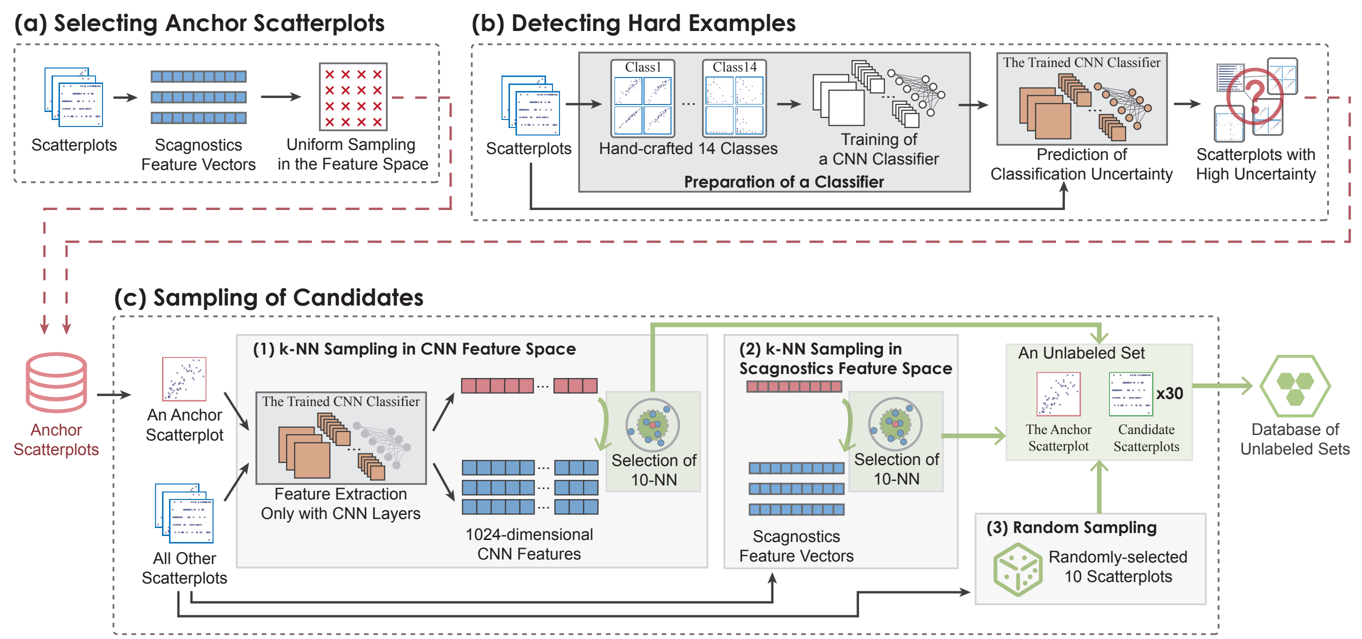


Fig. 2. The task generation procedure. (a) A set of anchor scatterplots is selected by uniform sampling in the feature space of Scagnostics. (b) A number of hard examples are filtered out as additional anchor scatterplots based on uncertainty sampling strategy. (c) Candidates are sampled by adopting three different sampling strategies.

Based on these principles, we have designed a generation procedure illustrated in Figure 2:

Selecting Anchor Scatterplots (Figure 2 (a)) To observe the principle of diversity, a uniform sampling was performed on the 50677 scatterplots by considering the distributions of 9 Scagnostics features.

Detecting Hard Examples (Figure 2 (b)) This step aims to address the second principle of uncertainty minimization. Motivated by the widely-used uncertainty sampling strategy in active learning [41], we design a classification-based strategy (Figure 2 (b)) for a preliminary coarse-grained uncertainty analysis as below:

1) First, we define fourteen classes of scatterplots according to the supplemental material of the user study in [8], where scatterplots are categorized into nearly non-overlapping classes based on visual perception.
2) Then, all the 50677 scatterplot images are manually classified into the fourteen classes and handled by a CNN classifier (Figure 2 (b)) with their class tags as a training dataset. The network structure (Figure 5 (a)) contains four convolution layers and two fully-connected layers.
3) Finally, the images are re-sent into the classifier to verify if their tags can be correctly predicted, and those that cannot be predicted as their assigned class tags are identified as hard examples and regarded as anchor scatterplots. The trained neural network layers in the blue box (Figure 5 (a)) with convolution layers can be considered as a feature extraction model and re-used in the next steps.

It should be pointed out that in this step the set of classes were derived from the controlled user study in [8] without any formal mathematical definition, which is still too difficult to cover all possible patterns in scatterplots. Using these summarized classes can effectively transform the hard example detection task into an uncertainty sampling problem, and to find as many easy-to-be-confused anchor scatterplots as possible. Indeed, the triplet labels tagged by annotators in the next stage play the key role of conveying visual recognition of similarities. As a result, the relatively general classes are sufficient for us to perform the detection task.

Sampling of Candidates (Figure 2 (c)) For each selected anchor image in the last two steps, we sample thirty candidates by accessing: 1) ten nearest neighbors from 1024-dimensional features extracted by the neural network layers mentioned above, 2) ten nearest neighbors from nine-dimensional Scagnostics features, and 3) ten randomly-selected ones from the rest of the scatterplots.

3.2 Judgment of Similarity

3.2.1 Labeling

The task in this stage is to judge similar scatterplot images and dissimilar images. We build a customized web system for collecting triplet labels. Figure 3 shows four panels in the interface: A) the anchor image, B) all thirty candidate images, C) the highlighted candidate image, and D) two lists of similar (blue stack) and dissimilar images (red stack). The annotator can drag an image from the candidate region to one of the lists, and a blue or red border will be added to the selected image to mark the selection. Additionally, the annotator can also browse the candidate images in the highlighted candidate region. Once all identifications are confirmed, the annotator submits the result and moves to the next set. If the annotator can not find similar or dissimilar images in an unlabeled set, the set can be skipped and replaced with another set.

For the labeling task, we recruited twenty-two annotators with undergraduate knowledge of statistics and mathematics. Each of them was paid \$0.05 (or gift cards with the same value) per valid labeled set. In order to assure labeling quality and avoid sloppy work, the following strategies are employed:

- **No Prior Knowledge** Based on the study in [8], annotators were not told what the criteria are for judging similarities. This

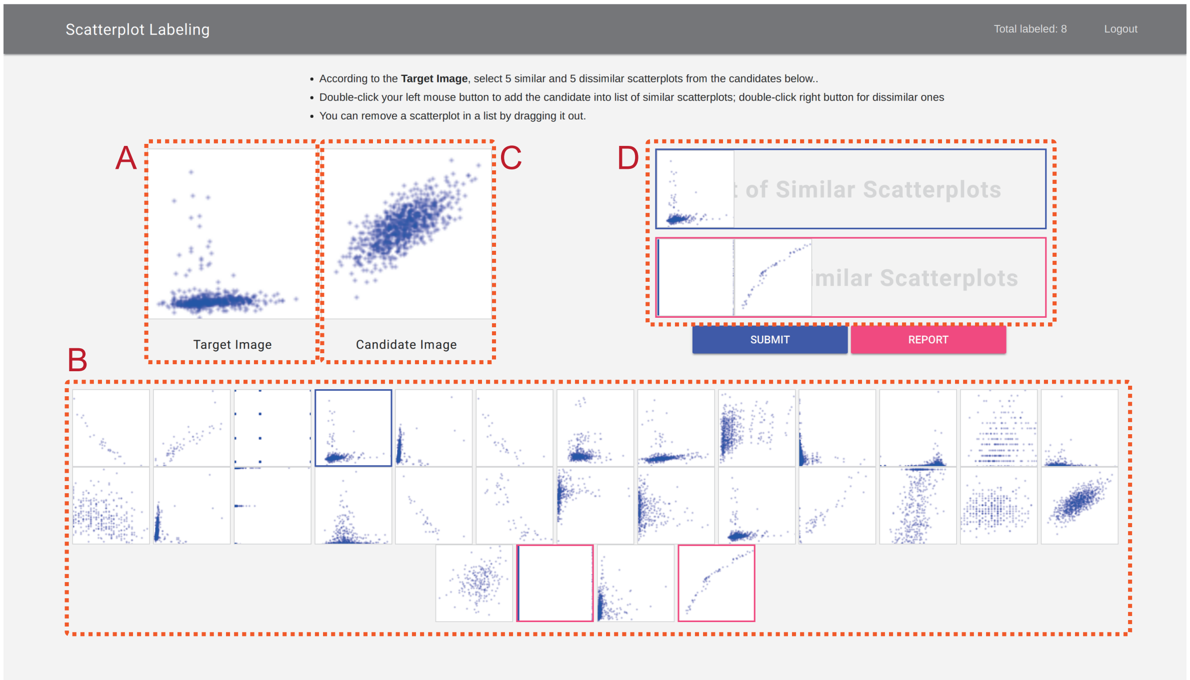


Fig. 3. The interface consists of four panels: (a) The anchor image, (b) 30 Candidate images, (c) Highlighted candidate, and (d) Lists of selected similar and dissimilar images.

strategy is to prevent subjective perception from being affected by pre-defined patterns or unrelated domain knowledge.
- **Task Redundancy** Each unlabeled set was distributed to three annotators. For the positive images, they are selected as training data in the model building stage only if it appears in all three annotators' positive lists.
- **Pre-labeled Test Sets** To test the labeling confidence of the annotators, we carefully designed ten unlabeled sets that were labeled by authors with obvious positive and negative candidates. Each annotator receives at least three test sets. If an annotator fails to identify those explicitly-arranged candidates, the labels from the annotator are manually investigated by us after the entire task.

3.2.2 Preliminary Analysis of Labeling Results

In the labeling stage, 5135 labeled sets were harvested. We perform a preliminary analysis between the labels and the origins of candidates. For each labeled set, we summarize:

- The number of positive scatterplots that are originally sampled based on the CNN features, Scagnostics features and random selection (i.e. method (1), (2) and (3) in Figure 2 (c)), respectively;
- The number of negative scatterplots that are not sampled based on the three methods.

These statistics are used for computing the intersections of human-assigned positive scatterplots and those which are close to the target scatterplot in the CNN and Scagnostics feature space, and vice versa for negative ones. Figure 4 shows the distributions of two measures on all labeled sets. In the two histograms, the distributions of CNN features and Scagnostics features are approximately correlated. Furthermore, the randomly-selected candidates rarely appear in the positive-scatterplot sets and always in negative ones. Specifically, in the histogram (a), more than 80% of the labeled sets have two or more scatterplots that do not come from

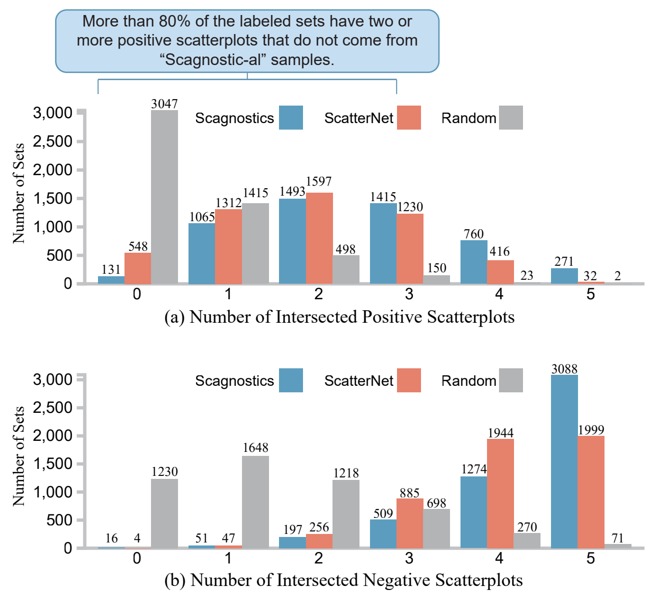


Fig. 4. Distribution of intersections between positive (a) / negative (b) scatterplots and candidates sampled based on the three candidate selection methods.

the "Scagnostic-al" samples, which indicates that the human annotators' choices were not consistent with the distances computed from the Scagnostics feature.

3.3 Building the Subjective Similarity Model

With the labeled triplets, our goal is to learn a non-linear embedding which represents human annotators' perception on similarity. Given two images I_1 and I_2, their Euclidean distance reflects how dissimilar I_1 and I_2 are. Because the features are extracted automatically, we choose triplet-based deep metric learning methods

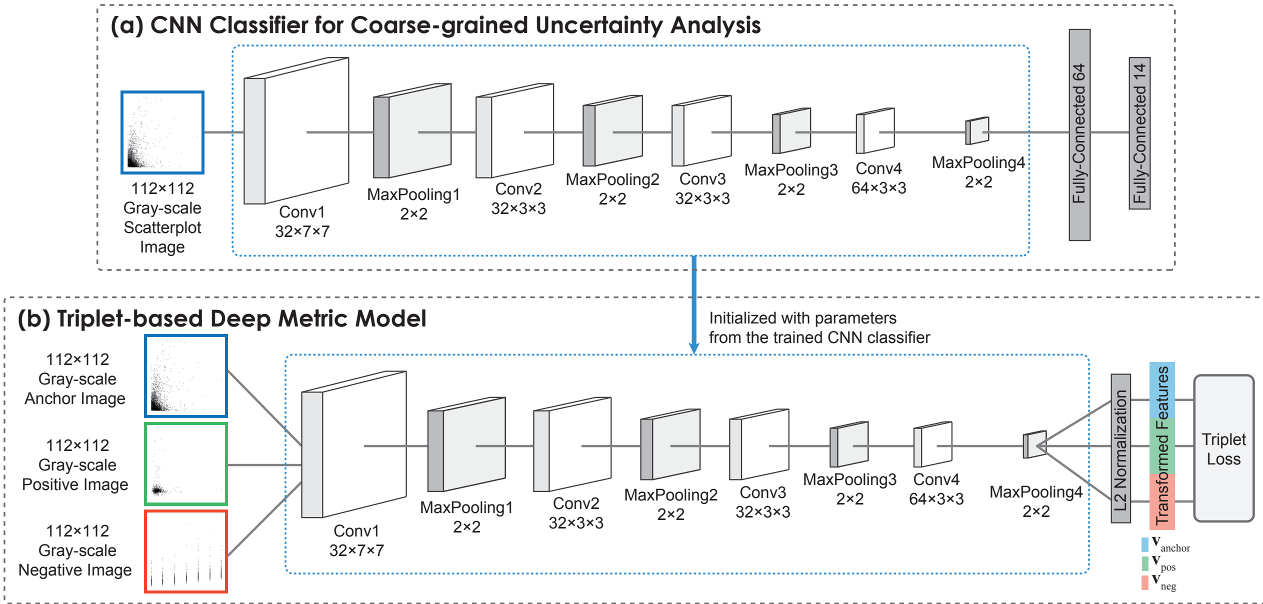


Fig. 5. The network structure of (a) the coarse-grained CNN classifier and (b) the triplet-based deep metric model. It should be noted that the weights of the convolution layers in the deep metric model are initialized with corresponding layers pre-trained in the step of detecting hard examples.

with CNN layers to transform the scatterplot images into the target feature space. This triplet network structure is widely used in the field of computer vision for constructing similarities among images [40], [42], [43]. Furthermore, the multi-layer CNN structure has been proven to gain significant performance on natural image classification [44] and object detection [45]. Specifically, the combination of convolution-pooling layers is a very typical network structure [44] with excellent performance in terms of feature extraction. In our method, the scatterplot images are used for training. Thus, the CNN structure is designed for extracting visual perception information hidden in the similarity labels.

The structure of ScatterNet is illustrated in Figure 5(b) where the convolution layers are initialized with the pre-trained CNN classifier described in Section 3.1.2. First, as each anchor scatterplot is associated with five positive and five negative scatterplots, the labeled sets are further transformed into triplets by combining the anchor scatterplot with one positive and one negative scatterplot. Thus, a single labeled set can be extended to $5 \times 5 = 25$ triplets. Three images in a training triplet are then fed into corresponding input layers. The input images pass an identical set of convolution layers in order to be transformed into embedded feature vectors, noted as $\mathbf{v}_{anchor}$, $\mathbf{v}_{pos}$ and $\mathbf{v}_{neg}$. It should be pointed out that the feature vectors are normalized to prevent results being affected by scaling. Finally we use the triplet loss function [46] defined in Equation 1:

$$\begin{aligned} Loss_{triplet}(\mathbf{v}_{anchor}, \mathbf{v}_{pos}, \mathbf{v}_{neg}) = \\ \max(0, \|\mathbf{v}_{anchor} - \mathbf{v}_{pos}\|_2^2 + \alpha - \|\mathbf{v}_{anchor} - \mathbf{v}_{neg}\|_2^2) \end{aligned} \tag{1}$$

where the hyper-parameter α controls the smallest margin in the embedded feature space.

This loss function constrains the distance relations among three images by giving penalty to the wrongly-embedded triplets where the distance between the anchor image and the positive one with margin α is still smaller than the distance from the anchor image to the negative one. The non-zero loss is back-propagated to all the CNN layers to update the corresponding parameters.

After the deep metric model is trained, the CNN part is separated for feature transformation. We pass an unlabeled scatterplot image through the CNN layers to get its embedding. The distance between two unlabeled images is defined as the Euclidean distance of the two corresponding embeddings.

3.4 Implementation Details

For the scatterplot images, we used Matplotlib[3] to generate the plots. To facilitate annotators' remote access of the labeling system, we chose server-browser architecture with MongoDB as the database management system, Python Flask for the server backend, and jQuery in the web browser. In the two deep-learning-related steps described in Section 3.1.1 and Section 3.3, the models can be easily implemented with most of the modern deep learning frameworks such as Tensorflow, Keras and PyTorch. In this work, Keras[4] with Theano backend was employed for the basic framework of the deep learning models.

4 Evaluation

We perform quantitative and qualitative experiments as well as two user studies to demonstrate effectiveness and efficiency of the trained model.

4.1 Test Dataset and Environment

The test dataset is collected from [8] for both experiments and the user study. It contains 247 different scatterplots carefully selected from a broad range of real-world datasets that contain various unique patterns. The variety of scatterplot patterns in the test dataset is essential for evaluating the efficacy of the proposed model for representing perceptual-oriented similarities. To avoid overfitting and affecting subsequent evaluations, all triplets that

3. http://matplotlib.org
4. https://keras.io

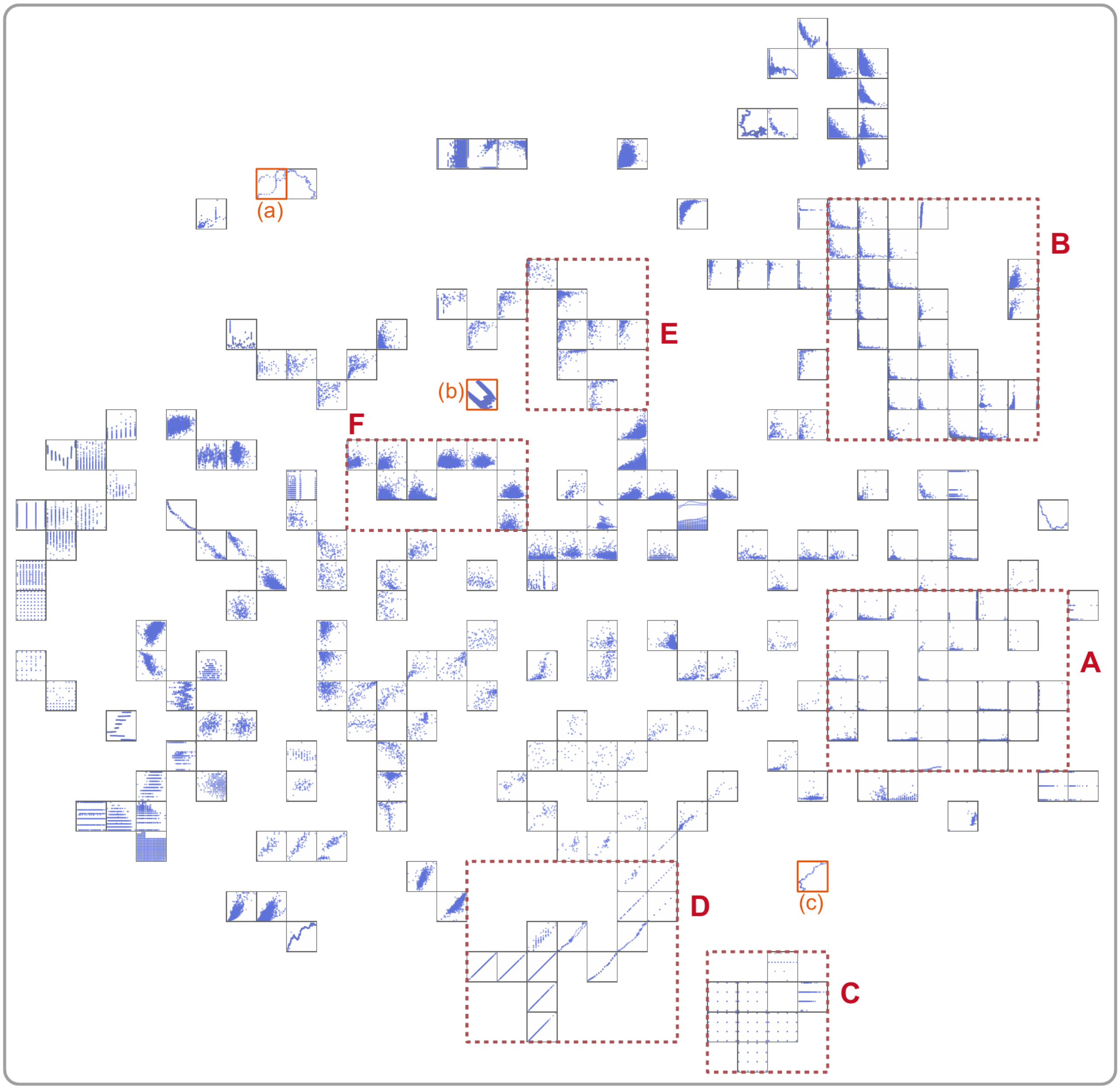


Fig. 6. The 2-D t-SNE projection result [47] of transformed feature vectors. The scatterplot images are aligned to grid. Six regions (from A to F) present several clusters that contain visually similar scatterplots.

contain scatterplots from the test dataset are excluded from training deep learning models. Because all duplicated data columns were removed (see Section 3.1.1), it can be guaranteed that there are no other scatterplots which are the same as the ones in the test dataset.

The training of deep neural networks was performed on a workstation with an Intel Xeon E3-1270v2 CPU and NVIDIA Quadro M4000 GPUs. For a single model it spent about 32 hours on a single GPU to finish 500 epochs. Each epoch consisted of about 80,000 triplets with the mini-batch size of 50.

4.2 Experiments

The goal of these quantitative experiments is to verify the training and testing performance of ScatterNet. In addition, a comparison with existing methods is conducted to evaluate how perceptual information is involved in different methods. Besides Scagnostics which is mentioned throughout the paper, we apply Histogram of Oriented Gradients (HOG) as another comparative method.

4.2.1 *Visualizing Embedding and k-NN Queries*

Figure 6 depicts a 2-D t-SNE (t-distributed stochastic neighbor embedding) projection [47] of transformed feature vectors from the trained model. To reduce visual clutter in the detail views, the scatterplot images are arranged in a grid form. From the perspective of visual perception, the following characteristics can be achieved from the result:

- Globally, the transitions of patterns in scatterplots are smooth and do not show very significant leaping, which is different from the Scagnostics method.
- Scatterplots that share similar patterns are noticeably grouped into small clusters. The red and dashed regions from A to F show similar scatterplots in Figure 6, such as group B

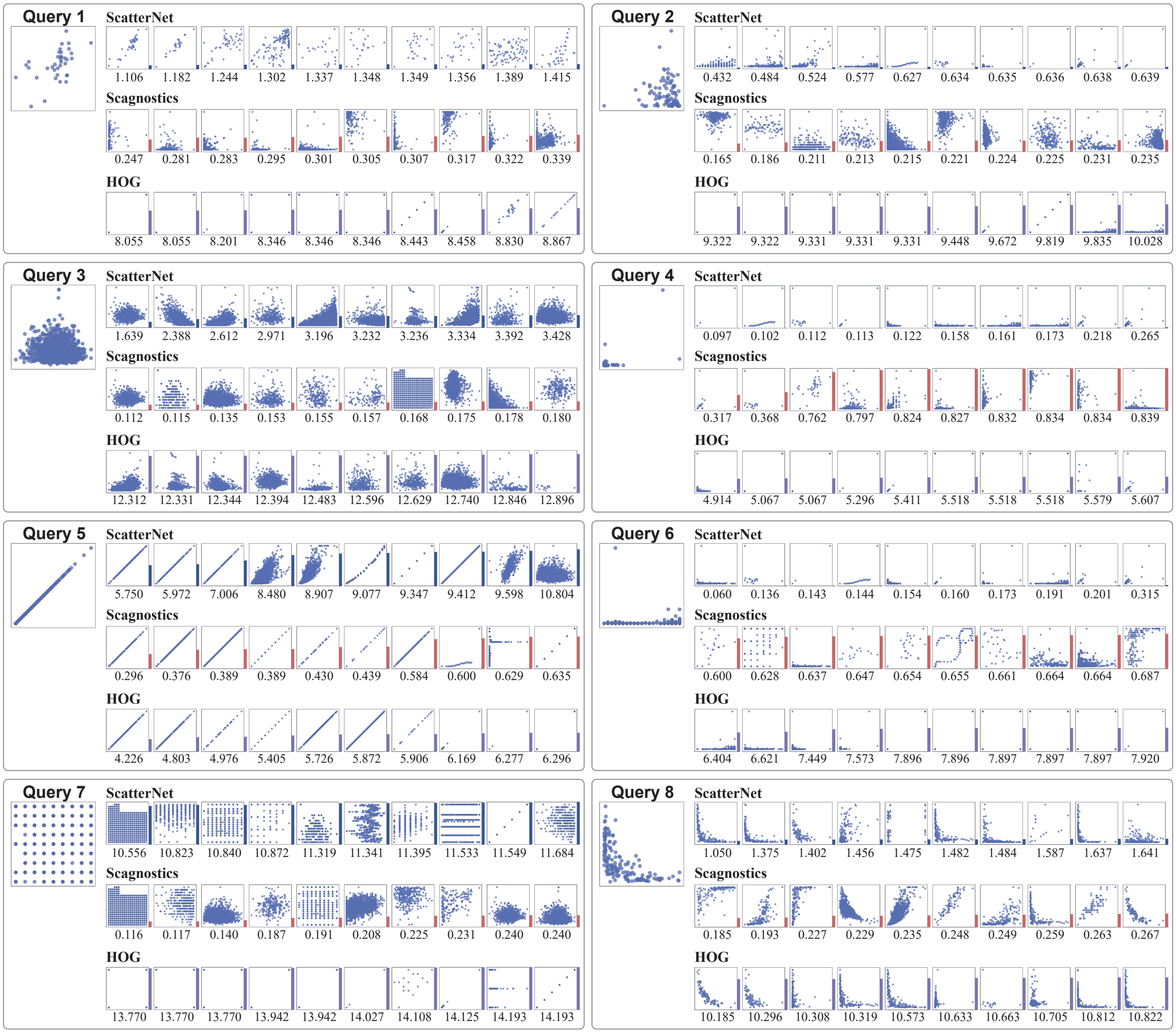


Fig. 7. Ten nearest neighbors of eight scatterplot queries based on ScatterNet, Scagnostics features and HOG features. The queries and k-NN results are all from the test dataset.

with long-tail distributions, group D with 45-degree linear correlation patterns, and group F with dense clusters.

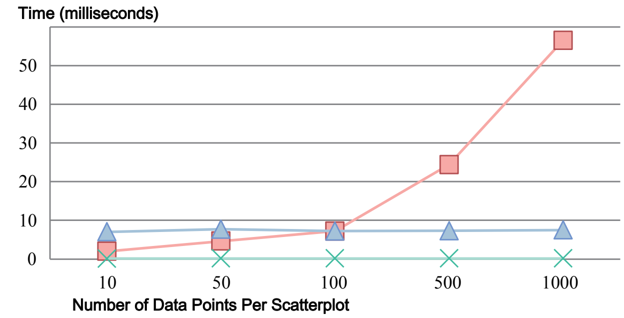
- There are several outlying scatterplots marked in small orange frames that do not belong to nearby groups, e.g., (a), (b), and (c) in Figure 6.

To evaluate the capability of embedding visually similar scatterplots, we compare ScatterNet with Scagnostics and HOG on eight representative categories of scatterplots from [8], as shown in Figure 7. The ranked list of ten nearest neighbors in ScatterNet and the corresponding ten most similar scatterplots in Scagnostics and HOG are placed upwards and downwards. The actual distance values from neighbors to the query are shown under the corresponding scatterplots and mapped to bars at the right side. Note that the heights of bars are scaled into $[0, MaxValue]$ respectively in the three methods; thus the distances from the same method can be compared. In Query 1, 2, 4, and 6~8, ScatterNet outperforms the other two methods regarding point distributions, geometric shapes, and density, while in Query 3 three methods are comparable. In the result sequence in Query 5, ScatterNet tends to return less-optimal results earlier than Scagnostics and HOG. This issue is also described in the first user study and further investigated in Section 5.2. From the perspective of distance distributions, it can be found that in all the three methods the distances increase drastically for dissimilar scatterplots, while for similar ones the distributions remain low and constant. This phenomenon may indicate that the methods tend to compress similar scatterplots into condensed clusters and expel the clusters from each other.

4.2.2 Performance Analysis

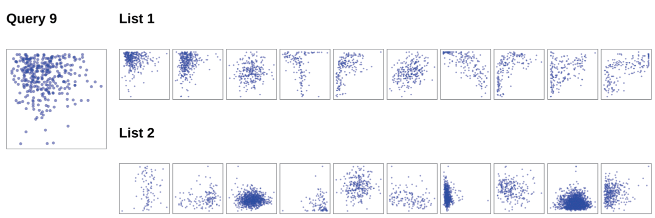
We compared the running performance of ScatterNet with Scagnostics on scatterplots with data points of different numbers in Figure 8. A Java implementation of Scagnostics[5] and Keras with Theano backend (both CPU and GPU) for ScatterNet is applied in this comparison. By using images as training sets, the transformation time from a scatterplot image to its feature

5. https://github.com/cran/scagnostics

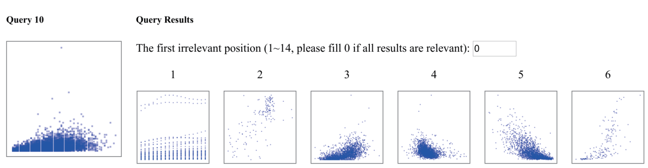


	10 Points	50 Points	100 Points	500 Points	1000 Points
Scagnostics	2ms	4.6ms	7.2ms	24.4ms	56.5ms
Our Model (CPU)	7ms	7.7ms	7.2ms	7.3ms	7.5ms
Our Model (GPU)	0.1ms	0.14ms	0.14ms	0.14ms	0.14ms

Fig. 8. Performance comparison between Scagnostics and ScatterNet for scatterplots with different number of data points.



(a) Task 1



(b) Task 2

Fig. 9. The interface of the system used for the two tasks. Task 1: (Left) The query scatterplot, (Right) Two lists of *k*-NN query results by employing ScatterNet and Scagnostics respectively. Task 2: (Left) The query scatterplot, (Right) A list containing randomized 7-NN query results from ScatterNet and Scagnostics.

vector is constant once the neural network structure and model parameters are ready. However, the computational complexity of Scagnostics features is proportional to the number of data points in the scatterplot [7]. The comparison indicates that Scagnostics takes a longer time than ScatterNet when the number of points is larger than a threshold. Note that the deep neural network can be accelerated with GPUs that parallelize the convolution operations.

4.3 User Studies

In this section, we describe a user study that was designed to test if our trained scatterplot similarity model can preserve visual perception. The purpose of the first user study is intended to reflect a general comparison among Scagnostics, HOG, and ScatterNet. Additionally, we designed a ranking task in the second user study to investigate *k*-NN query quality.

4.3.1 Task 1: Overview of Query Results

Setup and Procedure The participants were 24 graduate students (8 females and 16 males, ages ranging from 22 to 30 years old) from the college of computer science and the school of mathematical sciences in our university. To ensure the participants had knowledge of statistics and statistical charts, we selected the ones who have taken courses including probability theory, mathematical statistics, or information visualization. A desktop computer was used with a 24-inch LCD monitor at the resolution of 1920×1080 and Google Chrome web browser. The average distance from the participants to the monitor is about 70 cm, and the derived visual angle is 4.5297. It should be noted that none of the participants has ever enrolled in the similarity labeling process described in Section 3.2. After the user study, each participant was offered a $3 voucher for use in the university retail store.

We implemented a web-based user study system as shown in Figure 9 (a) which contains a query scatterplot and two lists of *k*-NN query results. One of the lists is computed with ScatterNet while another one comes from Scagnostics or HOG. Queries are randomly selected for each participant from the dataset described in Section 4.1, and the corresponding query results are from the dataset as well. The scatterplots in a single list are ranked accordingly in decreasing order of similarities to the query. The order of two lists were randomized to anonymize the association between lists and methods. The participants were asked to evaluate the quality of two results and choose the description that best represents the quality of two query results from four options: "List 1 is better", "List 2 is better", "Both are good", and "Both are bad".

The task was performed by each participant individually. The entire process included two steps which take about 20 minutes:

1) **Instruction:** A description page was given to the participant to illustrate the task and interaction in the interface.
2) **Evaluation:** The participant was asked to perform 40 sets of the comparison tasks. Their choices were stored and then counted.

Analysis From the evaluation we obtained 960 responses for all scatterplots in the dataset where each scatterplot was covered at least twice by Scagnostics and HOG. We transformed the answers into selection frequencies with the following method:

- The questions were divided into two groups based on whether Scagnostics or HOG results were contained, i.e., questions of "ScatterNet vs. Scagnostics" and "ScatterNet vs. HOG".
- In each group, for questions where "List 1 is better" or "List 2 is better" is selected, the count of the corresponding method will be increased by 1. For "Both are good" and "Both are bad", the counts for both (or neither of the) methods will be increased.

The frequencies of selections are summarized in Table 3. Two chi-square tests of independence were performed on the two tables, respectively:

TABLE 3
The two contingency tables for two groups. The count of selection is listed for each method.

(a) ScatterNet vs. Scagnostics

Conditions	Good	Bad	Total
ScatterNet	314	152	466
Scagnostics	122	344	466
Total	436	496	932

(b) ScatterNet vs. HOG

Conditions	Good	Bad	Total
ScatterNet	340	150	490
HOG	156	334	490
Total	496	484	980

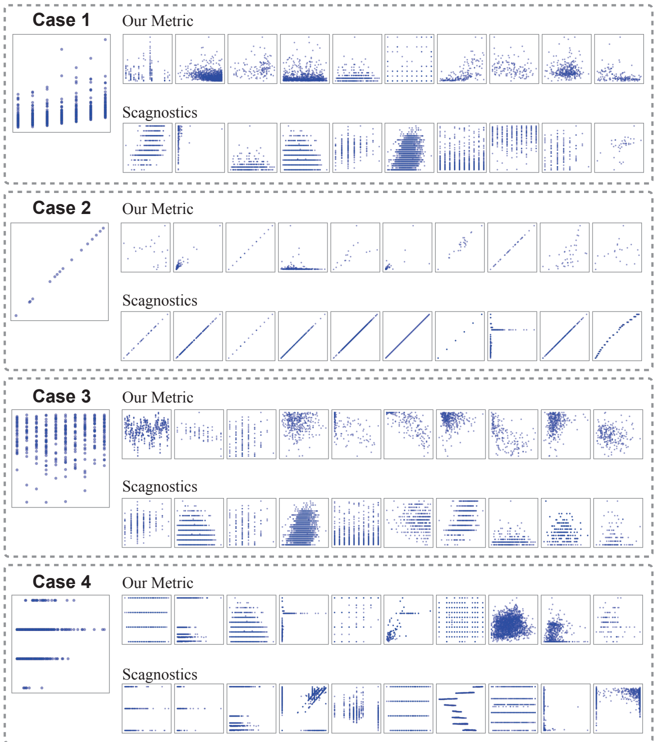


Fig. 10. Cases in task 1 with low ratings to ScatterNet where the query scatterplots share a common pattern of horizontal or vertical lines.

(a) ScatterNet vs. Scagnostics: $\chi^2(1,N=932)=162.06, p<0.001$
(b) ScatterNet vs. HOG: $\chi^2(1,N=980)=140.08, p<0.001$

which indicate significant differences in proportions.

To identify low-quality query results, we further explore query scatterplots which received no positive ratings in ScatterNet. In Figure 10, four representative cases are listed in which ScatterNet is not selected as the best results. We discover that the cases share a common characteristic of high "striated" value in their Scagnostics features, which presents a phenomenon that, for such type of scatterplots with many horizontal or vertical lines, Scagnostics features perform well or even better than ScatterNet. This issue may be due to the high efficacy of striated descriptor specially-defined to detect line patterns in Scagnostics. A similar issue was also reported in [8] where strong correlation is shown between similarities on the "striated" feature and visually-perceived similarities. This issue is further discussed in Section 5.2 from the perspective of attention mechanism of CNN layers.

4.3.2 Task 2: Ranking Scatterplots from Two Query Results

Setup and Procedure Another 16 participants were recruited in this task with the same requirement and reward as the first task. The test environment was the same.

In this task, we designed another web system as shown in Figure 9, where fourteen scatterplots are listed as query results with a specific query scatterplot placed on the left side. The scatterplot list consists of two 7-NN query results computed with ScatterNet, and another 7-NN results from Scagnostics or HOG. The initial orders and sources of the query results are anonymized and randomized. Each participant was asked to order these fourteen scatterplots. In the interface, participants can use mouse to drag a scatterplot images to the desired place in the list. For some query results, the similarity may be too small to be accepted as "similar scatterplots". Thus, we put an input field of "the first irrelevant position" in the interface for each ranking task. Participants can identify the first position where the results became insignificant to the corresponding query.

The entire task was performed individually within 40 minutes:

1) **Instruction:** A description page was given to the participant to illustrate the task and interaction in the interface.
2) **Evaluation:** The participant was asked to perform 30 sets of the ranking task. The ranking results were saved and uploaded.

Analysis By excluding the cases where the first irrelevant positions were set to 1, i.e., none of the scatterplots in the query result list was considered as a similar one, 421 ranking results were collected with 212 of "ScatterNet vs. Scagnostics" questions and 209 of "ScatterNet vs. HOG" ones. Similar to the grouping method used in Task 1, the questions were further divided into two groups, and results in the two groups were analyzed respectively.

In each group, the scores of two corresponding methods were assigned by performing the following processes:

- For ranked query results before the irrelevant position in each task, the scores were assigned as their order numbers.
- For query results at and after the irrelevant position, the scores were set as their average number of orders.
- Scores were summed up based on their corresponding sources of methods (ScatterNet, Scagnostics, or HOG). The lower score a method achieves, the better query results the method provides.

Among all 212 results in the group of "ScatterNet vs. Scagnostics", there were 156 results (73.58%) that ScatterNet receives a lower score than Scagnostics. The corresponding ratio in "ScatterNet vs. HOG" was 63.16% (132 out of 209). We further performed the Wilcoxon signed-rank test on the two lists of summed scores in each group. The results are listed below:

(a) ScatterNet vs. Scagnostics: $median(\text{ScatterNet})=46.0$, $median(\text{Scagnostics})=56.0$, $Z=7044.0$, $p<0.001$
(b) ScatterNet vs. HOG: $median(\text{ScatterNet})=48.5$, $median(\text{HOG})=52.5$, $Z=7367.5$, $p<0.001$

According to the results, in both of the groups the scores were significantly less for ScatterNet than for another one, which indicates that ScatterNet received better rankings.

5 Discussions

5.1 Adaptability Issue

As presented in Section 3.3, converting the input data into images is identical to visually representing the dots. To test whether the predictability of the model is stable when the visual encoding is changed, we applied the trained model as a baseline model (2px in radius and 0.4 for opacity) for evaluating scatterplots with different sizes (1px and 5px) and opacity values (0.2 and 0.6). In each set we computed the Jaccard indices of 10-NN query results between the baseline visual encoding and other visual encodings. Figure 11 reports the distribution of Jaccard indices. It can be seen that large numbers of the Jaccard index values are distributed in the range of 0 to 0.5, indicating that the visual encoding has an influence on the k-NN query results of ScatterNet. A feasible solution for reducing the influence is to adopt transfer learning strategy [48] in adapting ScatterNet to various visual attributes with a small quantity of new training data.

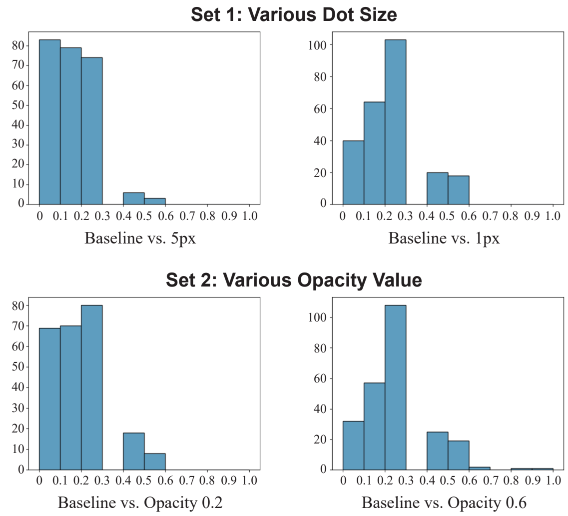


Fig. 11. Distributions of Jaccard indices for 10-NN query results between the baseline visual encoding and four others.

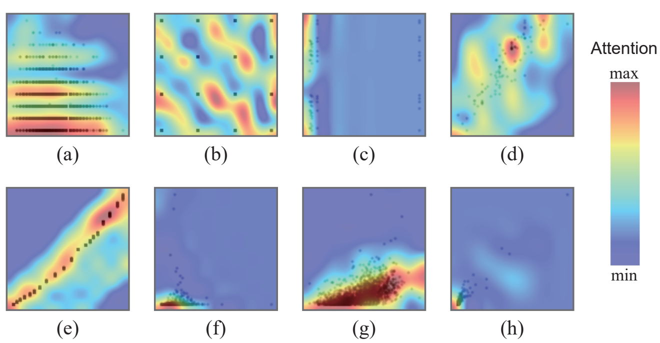


Fig. 12. Attention maps of eight different types of scatterplots. Attention value are visualized as heatmaps.

Another related issue is to build task-dependent similarity models. Because the standards for deciding similar and dissimilar scatterplots vary from task to task, re-labeled training data is necessary to be collected from domain experts. By using the baseline model as a pre-trained initialization, the training cost may be significantly reduced.

5.2 Inspection of the Trained Model

Unlike other interpretable models such as decision trees and linear regression, deep neural networks are commonly regarded as black-box models with low interpretability. In this section, we tend to investigate the internal mechanism of how ScatterNet recognizes scatterplots. Here we employ the Grad-CAM method [49] to generate attention maps of the input scatterplots. The attention map is utilized for depicting local regions that the convolution layers focus on when performing feature transformations, which implies the salience among different regions. Figure 12 shows a set of scatterplots in the test dataset blended with their corresponding attention maps. The attention maps are generated by using the last convolution layer in ScatterNet. From the results, we discover some interesting insights revealed by the attention maps and the scatterplots.

- **Global Patterns vs. Details**: The distribution of high-attention areas is related to the density of dots in the scatterplot. This indicates that the model prefers regions with high information density when judging similarities. Meanwhile, the regions with sparse points may be ignored by the model, meaning that the model tends to capture global patterns in the entire scatterplot image but not only local details.
- **Confusion on line patterns:** In the first task of the user study, it is discovered that the performance on scatterplots with multiple line patterns is not as good as those on other patterns. The situation can be partly presented in Figure 12 (a) where the high-value attention regions are not evenly distributed along the line patterns. A similar distribution can be achieved in Figure 12 (b) as well. This may be caused by the preference of global patterns mentioned above, i.e., the model regards multiple separated lines as a single pattern. A possible solution for this is to combine specifically-designed line pattern detectors into the deep neural network model.

5.3 Potential Use Cases

5.3.1 *Visual Querying by Scatterplot Images*

Similar to the example introduced in Section 1, it is a common task to retrieve desired patterns from a large collection of scatterplots for many applications such as Online Analysis Processing (OLAP). We develop an image search prototype to enable active querying of scatterplots. The interface (Figure 13) consists of two main components: a.1) a query field, and a.2) a result view. The query image is transformed into a feature vector by the trained similarity model, whose similarities to feature vectors in the database are measured with the Euclidean distance. Thereafter a list of similar scatterplot images is ranked in the result view. As an extension of querying specification, a query-by-sketching method [27], [50], [51], [52] can be employed to support freestyle drawing of target patterns on a canvas, which is then converted into images for searching.

5.3.2 *Visual Exploration of Subspaces in High-dimensional Data*

Visual exploration of massive scatterplots [15], [16], [53], [54] is an effective way for studying high-dimensional data. Subspace analysis is a common task to discover patterns in informative or task-related dimensions. One representative for visual subspace analysis is multi-dimensional projection that embeds data instances in a subspace into a 2-D scatterplot. When a dataset contains hundreds or thousands of dimensions, the pairwise combination of dimensions results in vast quantities of scatterplots. Thus, an automated pattern extraction method can be advantageous in detecting interesting patterns.

Usually, similarity measuring methods are deeply rooted in the foundations of such pattern detection algorithms, for instance, cluster analysis [22], [55], [56], [57], [58] and outlier detection [59], [60], [61]. As shown in Figure 13 (b), we design a prototype to support visual exploration and summarization of large-scale subspace projections. In the main view (b.1), the scatterplots of corresponding subspace are initially projected in accordance with their perceptual similarities derived from ScatterNet. The users are able to discover clusters and outliers of these scatterplots to find subspaces with similar or unique data distributions. By clicking on a scatterplot, the users can select or remove attributes in order to change the represented subspace of the scatterplot in the attribute selection view (b.2). Additional annotation means in Figure 13 (b.3) such as textual labels and cluster markers are provided to facilitate recording their discovered patterns.

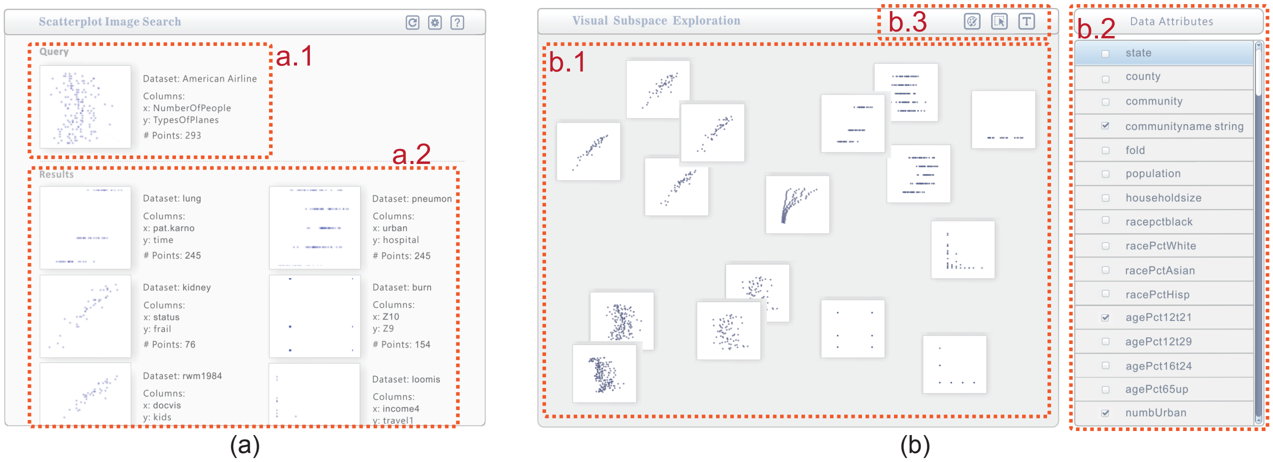


Fig. 13. Two use cases of ScatterNet. (a) The interface of the image search engine. A user can upload an image in the query field (a.1), and the search results are listed below in the result view (a.2). (b) Visual exploration of subspace projection scatterplots: (b.1) Main view, (b.2) Attribute selection view, and (b.3) Annotation view.

5.4 Limitations

There are some limitations in the data annotation and model training stages. Deep learning models usually demand a large amount of labeled data to achieve desirable prediction accuracy and generalization ability. Thus, training triplets requires massive time cost on data labeling. Unlike decision trees or support vector machines, a deep neural network often takes hours or even days to fit enormous numbers of training data instances, hence it is hard to provide interaction-level response for the user. In specific scenarios where the users want to re-annotate triplets based on analytical tasks or domain knowledge, the process could be time-consuming. Concerning model parameters such as the number of convolution layers and corresponding kernel sizes, there might be extra effort for tuning these values when the visual encoding or analytical tasks are changed.

As discussed in Section 5.2, the sparsity of data instances in scatterplots is another issue that affects the performance of ScatterNet. The CNN layers transform scatterplots that contain few points into a small region in the vector space, making themselves indistinguishable from each other. Thus, ScatterNet works well when the point distributions of scatterplots for training and prediction are relatively dense.

5.5 Future Work

In the future, we would like to further investigate the following issues. One promising extension of our work is to consolidate the user evaluation of our method by using rating scales. In the first user study, we mainly focus on comparing ScatterNet with two existing methods. To inspect the matching level between similarities perceived by human users and the computed results, rating scales like Likert scale can be utilized. Another extension of our work is to capture asymmetric similarity in scatterplots. In classical cognitive psychology, asymmetries exist in similarity judgment tasks [62]. As described in Section 3.3, currently we use Euclidean distance as the optimization target for the feature extraction layers in the deep neural network. However, asymmetry cannot be handled by metric distances such as Euclidean distance. A feasible solution is to design an alternative distance function that handles asymmetric distances well. The loss function should be modified as well to match the optimization goal.

The combination of visualization, perception, and computer vision raises exciting research trends. For defining similarities, semantic information, and domain knowledge can be involved in users' annotations of how similar two scatterplots are, which may lead to completely different presentations of similarity measures. Furthermore, by regarding visualization results as images, is it feasible that computer vision methods can assist in recognizing interesting regions? In our work we only take scatterplots into consideration. For other types of statistical charts or visual designs, is it possible to retrieve useful information from the images with deep learning methods to partly facilitate users' repetitive work on visual exploration and investigation? These challenges will have a positive influence on the design of automated information retrieval approaches in visualization results.

6 Conclusion

In this paper we propose a novel approach for characterizing perceptual similarities between scatterplots. Motivated by the success of deep neural networks in image recognition, we design a user-oriented data annotation stage to generate a set of triplets that convey human perception on the similarity of scatterplots. It is then used for training ScatterNet, a deep learning model to capture the similarity from scatterplot images. We carry out quantitative experiments and user studies on our trained model as well as a comparison with existing measures. The result indicates that ScatterNet outperforms existing solutions.

Acknowledgments

This work is supported by National 973 Program of China (2015CB352503), National Natural Science Foundation of China (61772456 and 61761136020), Key National Natural Science Foundation of China (61332017), National Key R&D Program of China (2017YFB1002803 and 2018YFB0904503), the National Research Foundation, Prime Ministers Office, Singapore under its Strategic Capability Research Centres Funding Initiative (N-CRiPT) and International Research Centre Funding Initiative (SeSaMe Center), and Singapore Ministry of Education Academic Research Fund Tier 1. We thank the authors of [8] for sharing the test dataset. The first author would like to thank a fawn for her helpful inspiration.

REFERENCES

[1] M. Friendly and D. Denis, "The early origins and development of the scatterplot," *Journal of the History of the Behavioral Sciences*, vol. 41, no. 2, pp. 103–130, 2005.

[2] E. Bertini, A. Tatu, and D. A. Keim, "Quality metrics in high-dimensional data visualization: An overview and systemization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 6, pp. 2203–2212, 2011.

[3] J. H. Friedman and W. Stuetzle, "John w. tukey's work on interactive graphics," *Annals of Statistics*, pp. 1629–1639, 2002.

[4] D. J. Lehmann, G. Albuquerque, M. Eisemann, M. Magnor, and H. Theisel, "Selecting coherent and relevant plots in large scatterplot matrices," *Computer Graphics Forum*, vol. 31, no. 6, pp. 1895–1908, 2012.

[5] L. Shao, T. Schleicher, M. Behrisch, T. Schreck, I. Sipiran, and D. A. Keim, "Guiding the exploration of scatter plot data using motif-based interest measures," *Journal of Visual Languages and Computing*, vol. 36, no. C, pp. 1–12, 2016.

[6] A. Tatu, G. Albuquerque, M. Eisemann, P. Bak, H. Theisel, M. Magnork, and D. A. Keim, "Automated analytical methods to support visual exploration of high-dimensional data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 5, pp. 584–597, May 2011.

[7] L. Wilkinson, A. Anand, and R. Grossman, "Graph-theoretic scagnostics," *Proceedings of IEEE Symposium on Information Visualization*, pp. 157–164, 2005.

[8] A. V. Pandey, J. Krause, C. Felix, J. Boy, and E. Bertini, "Towards understanding human similarity perception in the analysis of large sets of scatter plots," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2016, pp. 3659–3669.

[9] L. Harrison, F. Yang, S. Franconeri, and R. Chang, "Ranking visualizations of correlation using weber' s law," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1943–1952, 2014.

[10] M. Kay and J. Heer, "Beyond weber's law: A second look at ranking visualizations of correlation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 469–478, 2016.

[11] M. Sedlmair, A. Tatu, T. Munzner, and M. Tory, "A taxonomy of visual cluster separation factors," *Computer Graphics Forum*, vol. 31, no. 3.4, pp. 1335–1344, 2012.

[12] J. W. Tukey and P. A. Tukey, "Computer graphics and exploratory data analysis: An introduction," *The Collected Works of John W. Tukey: Graphics: 1965-1985*, vol. 5, pp. 419–432, 1988.

[13] P. A. Tukey, "Cognostics for looking at higher-dimensional data," in *Proceedings of The 18th Symposium on the Interface: Computer Science and Statistics*, 1986.

[14] L. Fu, "Implementation of three-dimensional scagnostics," *Masters Thesis, Department of Mathematics, University of Waterloo*, 2009.

[15] T. N. Dang and L. Wilkinson, "Scagexplorer: Exploring scatterplots by their scagnostics," *Proceedings of IEEE Pacific Visualization Symposium*, pp. 73–80, 2014.

[16] T. N. Dang, A. Anand, and L. Wilkinson, "Timeseer: Scagnostics for high-dimensional time series," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 3, pp. 470–483, 2013.

[17] T. N. Dang and L. Wilkinson, "Transforming scagnostics to reveal hidden features," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1624–1632, 2014.

[18] M. Behrisch, F. Korkmaz, L. Shao, and T. Schreck, "Feedback-driven interactive exploration of large multidimensional data supported by visual classifier," in *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2014, pp. 43–52.

[19] A. Anand and J. Talbot, "Automatic selection of partitioning variables for small multiple displays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 669–677, 2015.

[20] J. Seo and B. Shneiderman, "A rank-by-feature framework for interactive exploration of multidimensional data," *Information Visualization*, vol. 4, no. 2, pp. 96–113, 2005.

[21] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan, "Selecting good views of high-dimensional data using class consistency," *Computer Graphics Forum*, vol. 28, no. 3, pp. 831–838, 2009.

[22] J. Xia, W. Chen, H. Yumeng, H. Wanqi, H. Xinxin, and D. S. Ebert, "Dimscanner: A relation-based visual exploration approach towards data dimension inspection," in *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2016, pp. 81–90.

[23] A. Mojsilovic, J. Gomes, and B. E. Rogowitz, "Isee: Perceptual features for image library navigation," in *Proceedings of the SPIE, Human Vision and Electronic Imaging*, vol. 4662, 2002, pp. 266–277.

[24] A. Mojsilovic and B. E. Rogowitz, "Capturing image semantics with low-level descriptors," in *Proceedings of International Conference on Image Processing*, vol. 1, 2001, pp. 18–21 vol.1.

[25] B. E. Rogowitz, T. Frese, J. R. Smith, C. A. Bouman, and E. B. Kalin, "Perceptual image similarity experiments," in *Proceedings of SPIE, Human Vision and Electronic Imaging*, vol. 3299, 1998, pp. 576–590.

[26] A. Mojsilovic and B. E. Rogowitz, "Semantic metric for image library exploration," *IEEE Transactions on Multimedia*, vol. 6, no. 6, pp. 828–838, Dec 2004.

[27] L. Shao, M. Behrisch, T. Schreck, T. v. Landesberger, M. Scherer, S. Bremm, and D. A. Keim, "Guided sketching for visual search and exploration in large scatter plot spaces," in *EuroVis Workshop on Visual Analytics (EuroVA)*, 2014.

[28] A. Dasgupta and R. Kosara, "Pargnostics: Screen-space metrics for parallel coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1017–1026, 2010.

[29] M. Behrisch, B. Bach, M. Hund, M. Delz, L. von Ruden, J.-D. Fekete, and T. Schreck, "Magnostics: Image-based search of interesting matrix views for guided network exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 31–40, 2017.

[30] J. Schneidewind, M. Sips, and D. A. Keim, "Pixnostics: Towards measuring the value of visualization," in *Proceedings of IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2006, pp. 199–206.

[31] K. Reda, A. González, J. Leigh, and M. E. Papka, "Tell me what do you see: Detecting perceptually-separable visual patterns via clustering of image-space features in visualizations," in *Proceedings of IEEE Conference on Visual Analytics Science and Technology (VAST) (Poster)*, 2015, pp. 211–212.

[32] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer, "Revision: Automated classification, analysis and redesign of chart images," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, 2011, pp. 393–402.

[33] D. Jung, W. Kim, H. Song, J.-i. Hwang, B. Lee, B. Kim, and J. Seo, "Chartsense: Interactive data extraction from chart images," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2017, pp. 6706–6717.

[34] N. Siegel, Z. Horvitz, R. Levin, S. Divvala, and A. Farhadi, "Figureseer: Parsing result-figures in research papers," in *Proceedings of the European Conference on Computer Vision*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Springer International Publishing, 2016, pp. 664–680.

[35] S. Johansson and J. Johansson, "Interactive dimensionality reduction through user-defined combinations of quality metrics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 993–1000, 2009.

[36] R. A. Rensink and G. Baldridge, "The perception of correlation in scatterplots," *Computer Graphics Forum*, vol. 29, no. 3, pp. 1203–1210, 2010.

[37] R. A. Rensink, "The nature of correlation perception in scatterplots," *Psychonomic Bulletin & Review*, vol. 24, no. 3, pp. 776–797, 2017.

[38] G. Albuquerque, M. Eisemann, and M. Magnor, "Perception-based visual quality measures," in *Proceedings of IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2011, pp. 13–20.

[39] Ç. Demiralp, M. S. Bernstein, and J. Heer, "Learning perceptual kernels for visualization design," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 1, pp. 1933–1942, 2014.

[40] Y. Cui, F. Zhou, Y. Lin, and S. Belongie, "Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1153–1162.

[41] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, vol. 52, no. 55-66, p. 11, 2010.

[42] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

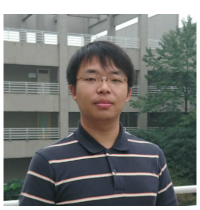
[43] X. Ji, W. Wang, M. Zhang, and Y. Yang, "Cross-domain image retrieval with attention modeling," in *Proceedings of the ACM International Conference on Multimedia*, 2017, pp. 1654–1662.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[45] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

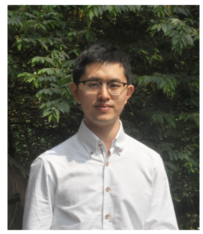
[46] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1386–1393.

[47] L. V. D. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.
[48] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Proceedings of Advances in Neural Information Processing Systems*, 2014, pp. 3320–3328.
[49] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conference on Computer Vision*, Oct 2017, pp. 618–626.
[50] M. Correll and M. Gleicher, “The semantics of sketch: A visual query system for time series data,” in *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2016, pp. 131–140.
[51] D. Schroeder and D. Keefe, “Visualization-by-sketching: An artist’s interface for creating multivariate time-varying data visualizations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 877–885, 2016.
[52] J. Walny, B. Lee, P. Johns, N. Henry Riche, and S. Carpendale, “Understanding pen and touch interaction for data exploration on interactive whiteboards,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2779–2788, 2012.
[53] J. Xia, F. Ye, W. Chen, Y. Wang, W. Chen, Y. Ma, and A. K. H. Tung, “Ldsscanner: Exploratory analysis of low-dimensional structures in high-dimensional datasets,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 236–245, Jan 2018.
[54] H. Liao, Y. Wu, L. Chen, and W. Chen, “Cluster-based visual abstraction for multivariate scatterplots,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 9, pp. 2531–2545, Sept 2018.
[55] M. S. Hossain, P. K. R. Ojili, C. Grimm, R. Muller, L. T. Watson, and N. Ramakrishnan, “Scatter/gather clustering: Flexibly incorporating user feedback to steer clustering results,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2829–2838, 2012.
[56] E. Packer, P. Bak, M. Nikkilä, V. Polishchuk, and H. J. Ship, “Visual analytics for spatial clustering: Using a heuristic approach for guided exploration.” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2179–88, 2013.
[57] T. Schultz and G. L. Kindlmann, “Open-box spectral clustering: Applications to medical image analysis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2100–2108, 2013.
[58] H. Chen, W. Chen, H. Mei, Z. Liu, K. Zhou, W. Chen, W. Gu, and K. L. Ma, “Visual abstraction and exploration of multi-class scatterplots,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1683–1692, Dec 2014.
[59] L. Cao, M. Wei, D. Yang, and E. A. Rundensteiner, “Online outlier exploration over large datasets,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 89–98.
[60] N. Cao, C. Shi, S. Lin, J. Lu, Y. R. Lin, and C. Y. Lin, “Targetvue: Visual analysis of anomalous user behaviors in online communication systems,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 280–289, 2016.
[61] J. Zhao, N. Cao, Z. Wen, Y. Song, Y.-r. Lin, and C. Collins, “#FluxFlow : Visual analysis of anomalous information spreading on social media,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 1, pp. 1773–1782, 2014.
[62] A. Tversky, “Features of similarity.” *Psychological Review*, vol. 84, no. 4, pp. 327–352, 1977.

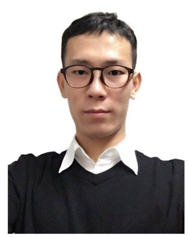


Yuxin Ma Yuxin Ma is a Ph.D. student in the State Key Lab of CAD&CG, Zhejiang University, China. His current research focuses on information visualization, visual analytics, and visual data mining.

Anthony K.H. Tung Anthony K.H. Tung received his B.S. (second class honor) and M.S. degrees in computer science from the National University of Singapore (NUS), in 1997 and 1998, respectively, and Ph.D. degree in computer science from Simon Fraser University in 2001. He is currently an associate professor in the Department of Computer Science, NUS. His research interests include various aspects of databases and data mining (KDD) including buffer management, frequent pattern discovery, spatial clustering, outlier detection, and classification analysis.



Wei Wang Dr. Wei Wang is an assistant professor in the Department of Computer Science, National University of Singapore (NUS). He got his B.S. from Renmin University of China in 2011 and PhD from NUS in 2017. His research interests include deep learning systems and applications for multimedia data. His papers were published in VLDB, ACM Multimedia, VLDBJ, TOMM. He has served as a PC member for ACM Multimedia, VLDB, ICDE and DASFAA.



Xiang Gao Xiang Gao is a master student in the Department of Computer Science in Hangzhou Normal University, China. His current research focuses are information visualization and virtual reality.

Zhigeng Pan He got his Ph.D degree in 1993 and became a full professor in Zhejiang University in 1996. He has published more than 100 technical papers on important journals and conferences including TPAMI, TVCG, IEEE Multimedia, ACM Multimedia and IEEE VR. He is a member of IEEE and ACM SIGGRAPH. His research interests include virtual reality, computer graphics and HCI. Currently, he is the Editor-in-Chief of Transactions on Edutainment. He is the program co-chair of CASA 2011, SIGGRAPH Asia 2011 (Sketches and Posters), IEEE VR 2013, SIGGRAPH Asia 2016(Symposium on Education), and the conference co-chair of VRCAI2012, VRCAI2013, VRCAI2015 and CW2016.

Wei Chen Dr. Wei Chen is a professor in the State Key Lab of CAD&CG, Zhejiang University. He has published more than 60 papers in international journals and conferences. He served as steering committee of IEEE Pacific Visualization, conference chair of IEEE Pacific Visualization 2015, and paper co-chair of IEEE Pacific Visualization 2013. For more information, please refer to http://www.cad.zju.edu.cn/home/chenwei/.