VISAtlas: An Image-Based Exploration and Query System for Large Visualization Collections via Neural Image Embedding

Yilin Ye, Rong Huang, and Wei Zeng, *Member, IEEE*

Abstract—High-quality visualization collections are beneficial for a variety of applications including visualization reference and data-driven visualization design. The visualization community has created many visualization collections, and developed interactive exploration systems for the collections. However, the systems are mainly based on extrinsic attributes like authors and publication years, whilst neglect intrinsic property (i.e., visual appearance) of visualizations, hindering visual comparison and query of visualization designs. This paper presents *VISAtlas*, an image-based approach empowered by neural image embedding, to facilitate exploration and query for visualization collections. To improve embedding accuracy, we create a comprehensive collection of synthetic and real-world visualizations, and use it to train a convolutional neural network (CNN) model with a triplet loss for taxonomical classification of visualizations. Next, we design a coordinated multiple view (CMV) system that enables multi-perspective exploration and design retrieval based on visualization embeddings. Specifically, we design a novel embedding overview that leverages contextual layout framework to preserve the context of the embedding vectors with the associated visualization taxonomies, and density plot and sampling techniques to address the overdrawing problem. We demonstrate in three case studies and one user study the effectiveness of *VISAtlas* in supporting comparative analysis of visualization collections, exploration of composite visualizations, and image-based retrieval of visualization designs. The studies reveal that real-world visualization collections (e.g., Beagle and VIS30K) better accord with the richness and diversity of visualization designs than synthetic collections (e.g., Data2Vis), inspiring composite visualizations are identified in real-world collections, and distinct design patterns exist in visualizations from different sources.

Index Terms—Visualization collection, image embedding, visual query, image visualization, design pattern

1 Introduction

Visualization is an interdisciplinary research discipline that covers a variety of data types, analytical tasks, and visual representations. The process of creating visualization is essentially a function that maps from data domain to imagery domain[15]. As such, there are obvious benefits of collecting and analyzing imagery collections of visualization. Visualization reference systems (e.g., [4], [56]) utilize visualization icons to help quickly understand sub-disciplines. Visualization researchers leverage fine-annotated visualization collections to build machine learning models for chart classification (e.g., [5], [44]), and to reveal the design space of data visualization (e.g., [16], [23]). Data-driven visualization design, including automatic chart generation and recommendation (e.g., [39], [76]), can also benefit from sophisticated design principles curated from high-quality visualization collections.

In line of this trend, many efforts have been recently devoted to constructing visualization collections, such as [5], [8], [14], [16], [24], [25], [54]. The visualization collections are inherently diverse in many perspectives: from different data sources, by different research groups, and in different manners. For example, Beagle[5] crawls visualizations from the web using keyword search, while VIS30K[14] extracts visualizations from IEEE VIS publications via convolutional neural networks (CNNs). Although many opportunities have been exploited, the diverse visualization collections also pose obstacles for intensive usage scenarios. Potential users may wonder: Which visualization collection shall be used? Are there any similar visualizations in the collection? What visual features are prominent in a visualization? What visual features are frequently composited in visualizations?

Answering these questions would require an effective exploration and query system for visualization collections. Existing systems for visualization images mined from scientific papers such as VIStory[73] and VISImageNavigator[14] are mainly based on extrinsic attributes including author names and publication years. The property hinders the generalization of these systems to other collections without the attributes. More importantly, attribute-based systems neglect the intrinsic property, i.e., visual appearance of visualizations that is also perceived by humans. This hinders vision-based tasks including visual comparison and query of visualization designs.

- *The authors are with the Hong Kong University of Science and Technology (Guangzhou), Guangzhou, Guangdong 511400, China, and also with the Hong Kong University of Science and Technology, Hong Kong SAR, China. E-mail: yyebd@connect.ust.hk, rhuang421@connect.hkust-gz.edu.cn, weizeng@ust.hk.*

Manuscript received 29 August 2022; revised 29 November 2022; accepted 6 December 2022. Date of publication 14 December 2022; date of current version 26 June 2024.
This work was supported by the National Natural Science Foundation of China under Grant 62172398.
(Corresponding author: Wei Zeng.)
Recommended for acceptance by P. Bremer.
This article has supplementary downloadable material available at https://doi.org/10.1109/TVCG.2022.3229023, provided by the authors.
Digital Object Identifier no. 10.1109/TVCG.2022.3229023

1077-2626 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.
orized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 28,2025 at 20:09:02 UTC from IEEE Xplore. Restrictions a

To fill this gap, we design *VISAtlas* – an image-based exploration and query system for visualization collections. VISAtlas consists of two main components. First, we build a neural image embedding model that converts raw visualization images to human-interpretable embedding vectors of visualization taxonomies (Section 3.1). To improve the embedding accuracy, we build a comprehensive dataset with both synthetic and real-world visualizations that cover the rich diversity of data visualization in aspects of chart types, data distribution, colormaps, and realistic usage (Section 4.1). We also leverage CNNs with a triplet loss to optimize embedding distance (Section 4.2).

Second, we design an interactive interface that supports exploration and query of visualization collections based on the embedding vectors. The interface is essentially a coordinated multiple view (CMV) system that enables the exploration of embedding vectors from multiple perspectives. The interface integrates a novel design of *Embedding Overview*, which uses contextual layout framework[18] to simultaneously present relations of the embeddings and relations of the embeddings to visualization types (Section 5.1). We further leverage density plot and sampling techniques to address the overdrawing problem. Besides, *Embedding Histogram* is used to reveal embedding distributions of each visualization type (Section 5.2). The interface is complemented with *Visualization Gallery* (Section 5.3) that presents similar visualizations with user-input ones specified by filtering and visual query interactions (Section 5.4).

We elaborate the utility and effectiveness of VISAtlas system via studies on comparing different visualization collections (Section 6.2), exploring composite visualization (Section 6.3), and retrieving visualization designs for both real and hand-drawn visualizations (Section 6.4). The studies reveal some interesting findings, such as real-world visualization collections (e.g., Beagle and VIS30K) better accord with the richness and diversity of visualization designs than synthetic one (e.g., Data2Vis), inspiring composite visualizations of *Matrix* and *Table* types are identified in real-world visualization collections, and distinct design patterns exist in visualizations from different sources.

Our work mainly makes the following contributions:

- We develop an effective neural image embedding method for visualization images. With comprehensive training dataset and CNN-based classification model with contrastive learning using triplet loss, the embedding model can accurately classify raw visualization images into human-interpretable visualization taxonomies and produce high-quality embedding representations for visualizations.
- We develop a web-based CMV system that facilitates multi-perspective exploration and visual query of visualization collections. Specifically, we propose a novel design that takes advantages of contextual layout framework for embedding projection, together with density plot and sampling for clutter reduction.
- We perform comparative analysis of different visualization collections and identify their differences in visualization richness and diversity. We also demonstrate how VISAtlas system can be exploited for exploring composite visualizations and retrieving design references in existing collections.

2 Related Work

Visualization Collection. A fundamental basis for data-driven visualization design is to build a suitable visualization collection that reflects the richness and diversity of visualization designs. Many studies (e.g., [25], [32], [33], [43], [72]) tackle the challenge with a twofold process: first, construct a raw dataset featuring diverse data fields (e.g., categorical, ordinal, high-dimensional, etc.); and second, map the raw data to visualizations using authoring libraries like ggplot [66], D3[10], and Vega-Lite[53]. Empirical design rules can be formulated as constraints to promote good visualizations [39], [76]. However, these datasets are typically limited to certain visualization types due to the expressivity of authoring tools. As an example, Data2Vis[25] only has four visualization types including point, line, area, and bar charts. Scientific and graph visualizations are missing. As such, the datasets are not suitable for tasks that cover complete visualization types.

Another approach to the problem is to crawl real-world visualizations. ReVision[54], MASSVIS[8], Beagle[5] draw a corpus of visualization images from the web. Interesting visualization usage patterns, such as line and bar charts are dominating on the web whilst pie charts are uncommon, are revealed[5]. Nevertheless, web-based visualizations are of varying quality that requires substantial efforts to filter out low-quality ones. Instead, some other researchers collected visualization images from publications in top visualization venues including IEEE TVCG and IEEE VIS conference (e.g., [14], [16], [24]). The visualization collections show great potential in identifying related work and understanding VIS[14], and revealing composition and configuration patterns in multiple-view visualizations[16].

In addition to visualization collections, existing works also provide interactive systems (e.g., [14], [57], [73]) for exploring visualizations, based on extra attributes like venues and authors. However, the systems omit visual appearance that is also perceived by humans[6]. To address the issue, some researchers focus on developing more sophisticated retrieval techniques based on visual features of visualizations. However, these techniques are mostly designed for visualizations with structural information, e.g., SVG [31], [38], Vega-Lite grammars[76], and infographics with fine-grained labels[50], [52]. Instead, this work focuses on visualizations in bitmap image format that is the most commonly available type of chart. The gap hinders potential applications including visual comparison of visualization subfileds[37] and visual query of visualization designs based on hand-drawn sketches[44]. We fill the gap with an image-based system for visualization collections, and demonstrate how the system can support the comparison of visualization collections (Section 6.2), identify composite visualizations (Section 6.3), and query based on user-input designs (Section 6.4).

Visualization for Image Collection. Visualizing large image collections is an active field of research in visual computing. Many visualization methods have been developed, which can be generally categorized as attribute- and image-based

orized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 28,2025 at 20:09:02 UTC from IEEE Xplore. Restrictions a

approaches. Attribute-based approaches organize images based on faceted metadata to facilitate search and browse [71]. Related systems typically provide an overview of the faceted metadata, coupled with progressive menus that allow users to look into one single image, e.g., [20], [51]. Spatial distribution can be utilized when the image datasets are coupled with geographical locations [58], [63]. Many visualization reference systems (e.g., [4], [56]) organize visualization images according to certain attributes like data fields and layout patterns. VIStory [73] and VISImageNavigator [14] also adopt the strategy by organizing visualization images according to certain attributes like publication venue, year and author.

On the other hand, image-based methods extract visual properties of images, and organize the images in certain layouts based on pairwise image similarities [48], [65]. The approach allows users to explore a visualization collection solely based on the rendered images, which has many benefits such as helping assess visualization quality [6], [36], comparing visual presentation designs between disciplines [27], and customizing dashboards from user sketches [44]. However, most image-based methods utilize image similarities measured upon low-level visual features (e.g., color and texture) that lack semantic information, which increases the difficulty for users to interpret the visualizations. Human interpretable semantics can be annotated by users [70] or described using descriptive captions via deep learning [69].

This work adds to the line of image-based visualization for image collections, with a particular focus on visualization collections. A core requirement here is to transform raw visualization images into human-interpretable embeddings. To fulfill the requirement, we leverage neural image embedding that describes a visualization image according to the visualization taxonomy (Section 3.1). We train a CNN model on a new dataset with synthetic and real-world visualizations (Section 4.1), and employ a triplet loss to improve the embedding accuracy (Section 4.2).

Image Embedding. Image embedding represents an image with a feature vector, which can be used to facilitate similarity measurement or clustering of images. Traditional methods for image embedding often require more explicitly defined feature extractors such as SIFT descriptors [41]. In contrast, neural image embeddings that represent images with a vector of numbers enable more flexible feature extraction. Off-the-shelf CNN extractors show superior performance to traditional feature extraction methods in many tasks. Notably, neural image embedding shows superior performance in chart classification [13] compared to traditional methods such as SVM used in ReVision [54]. ChartSense [35] exploits this technique to incorporate chart classification into an interactive system for chart analysis. ChartOCR [42] leverages neural-embedding-based chart classification combined with OCR methods for data extraction of three chart types. In addition, neural chart classification models are used in key intermediate step for visual encoding extraction [49] and figure parsing [59]. The visualization community has also employed neural image embedding to address other visualization-related tasks, such as to capture perception-driven similarities of scatterplots [45], to represent flow lines or surfaces [28], and to encode information in visualizations [74].

Image embeddings are high-dimensional data, which can be feasibly projected to a 2D space using dimension reduction algorithms such as PCA [68], t-SNE [64], UMAP [47] and MDS [21]. Most of the methods are designed to preserve pairwise distances between data points, preferable for clustering and abnormality detection tasks in applications like document exploration [7] and emoji analysis [40]. However, the methods often fail to preserve any possible explicit attribute information, which reduces the interpretability of latent space embedding vectors [19]. To achieve stable and meaningful dimension reduction, Cheng and Mueller [18] proposed a contextual layout framework that preserves the context of the data with the associated attributes. High-dimensional data points are projected to a radial layout where the position of each data point is calculated based on the balance of forces proportional to the values in each attribute [19], [30].

This work also projects image embeddings to a radial layout, to provide a common explainable 2D space for the overview and comparison of visualization collections. To improve scalability, we further employ density contour and sampling techniques to reduce overdrawings caused by large visualization collections (Section 5.1). Embedding histograms (Section 5.2) are also presented to provide multi-perspective information of the image embeddings.

3 Overview

This section discusses the choice of visualization taxonomy (Section 3.1) for neural image embedding. Next, we present the requirement analysis (Section 3.2), followed by an overview of our VISAtlas system (Section 3.3).

3.1 Visualization Taxonomy

This work aims to develop an image-based exploration and query system for large visualization collections. Existing image-based exploration systems for image collections (e.g., [48], [65]) typically treat images as high-dimensional vectors, which are then projected to a 2D plane using dimension reduction methods. However, directly projecting images to a 2D plane has several limitations. First, dimension reduction methods are typically constrained to images of fixed resolution (e.g., 256×256 in [65]), or image resizing that may affect the projection results is required. Instead, this work aims to support visualization images of any resolution. Second, the approach relies on low-level image similarity metrics such as spectrum distance and color histogram distance, which are vulnerable to various non-data visuals such as background color. More importantly, the metrics are not intuitive to humans, making the visualization less interpretable.

Instead of relying on low-level image features, we choose to encode visualization images to human interpretable semantics. To this end, we opt for a taxonomical classification of visualization images based on graphical encodings. Specifically, we adopt the taxonomy that classifies static visualization images based on the visual features of visualization images [8]. The taxonomy defines 12 primary visualization types, including *Area*, *Bar*, *Circle*, *Diagram*, *Distribution*, *Grid & Matrix*, *Line*, *Map*, *Point*, *Table*, *Text*, and *Trees & Networks*. Each primary type further includes a

orized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 28,2025 at 20:09:02 UTC from IEEE Xplore. Restrictions a

number of subtypes. We plot into the subtypes and notice that subtypes in *Distribution* are visually similar to some subtypes in other primary types, such as *dot array* in *Distribution* versus *dot plot* in *Point*, and *histogram* in *Distribution* versus *bar chart* in *Bar*. We decide to omit the *Distribution* type, and keep the remaining 11 types as the visualization taxonomy of our work.

3.2 System Requirements

Chen et al. [14] identified common use-cases for collections of visualization images, including *identifying related work*, *teaching and communication*, *understanding VIS*, and *tool building and testing*. They also built VISImageNavigator to view and query the VIS30K dataset based on metadata like keywords and authors. An *image-based* exploration and query system for visualization collections is complementary to the attribute-based method. It can benefit researchers and novices in visualization, as well as common users who want to apply visualization to their own tasks. For instance, researchers on building novel tools for visual question answering or visualization re-targeting need to evaluate and compare visualization collections as training data for machine learning algorithms. Moreover, visualization novices interested in understanding VIS require effective visual searching to browse visualization images and identify similar designs. In addition, common users outside the visualization community can also discover useful visualization design for their own applications, such as making figures for academic papers in other areas and visualizing business statistics in powerpoint.

In response to the needs of potential users, we form two research goals for the image-based exploration and query system for visualization collections. First, the system is expected to *support visual exploration and comparison of visualization collections (G1)*, to help researchers on machine learning for VIS identify suitable datasets. Second, the system is expected to *enable image-based query of visualization design (G2)*, to facilitate design recommendations for visualization designers. With this in mind, we refer to closely-related fields including image embedding, high-dimensional visualization, and data-driven design, and compile a set of system requirements, as follows:

R1: *Effective image embedding*. Image embedding is chosen to convert raw visualization images to human interpretable semantics. Both *G1* and *G2* require an effective embedding method for visualization images. Here, embedding vectors should conform to the correct taxonomical classification of visualization type (**R1.1**), e.g., a bar chart should be primarily classified as *Bar* type but not *Circle* type, while a pie chart should be primarily classified as *Circle* type but not *Bar* type. Second, embedding vectors should reflect the visual similarity of raw visualizations, which forms the basis for displaying image data according to class similarity (**R1.2**), i.e., visually similar visualizations are expected to have smaller distance than dissimilar ones. For instance, the distance between embedding vectors of two bar charts shall be smaller than that of a bar chart and a pie chart.

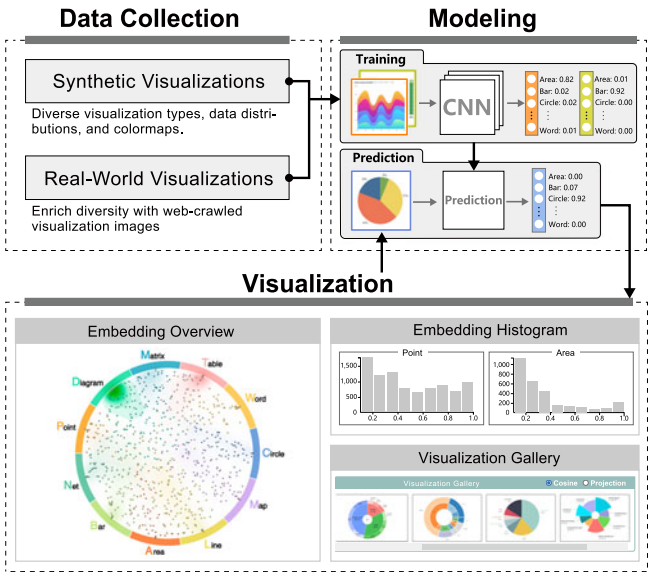


Fig. 1. Overview of VISAtlas workflow. Our system consists of three stages: Data collection, modeling, and visualization.

R2: *Intuitive visual representation*. The embedding vectors are high-dimensional data that require intuitive visual representation to support *G1*. For exploring the embedding information of single chart and comparing the similarity between visualizations, the system is required to simultaneously present 1) embedding vectors in respect to each visualization type, and 2) the distance of embedding vectors for different visualizations (**R2.1**). Second, overdrawing problems in visual representation caused by excessive data in large visualization collections need to be solved (**R2.2**). Last, multi-faceted information presentation is required to explore the embedding vectors from multiple perspectives (**R2.3**).

R3: *Efficient user interaction*. The system needs efficient user interaction to support both visual exploration (*G1*) and query (*G2*) of visualizations. Specifically, user-friendly interactions are needed to facilitate the exploration of visualization collections and help identify visualizations with user-specified attributes (**R3.1**). Moreover, the system also needs to support image-based query of visualizations that are visually similar to user-input ones (**R3.2**).

3.3 System Overview

We develop a web-based system, namely *VISAtlas*, that fulfills the system requirements. As illustrated in Fig. 1, the system mainly consists of three stages: 1) *data collection*, 2) *modeling*, and 3) *visualization*. In the *data collection* stage, we construct a new collection of visualization images with synthetic and real-world visualizations (Section 4.1). Here we carefully consider diverse visualization types, data distributions, and colormaps, when synthesizing visualizations. We further enrich the diversity of the visualization collection with real-world visualizations crawled from the web, which facilitates the classification results as shown by experiment results (Section 6.1).

Next, in the *modeling* stage, the visualization images are fed into a convolutional neural network (CNN), yielding an

orized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 28,2025 at 20:09:02 UTC from IEEE Xplore. Restrictions a

embedding model for visualization images (Section 4.2). We annotate each image with the label of visualization type, and train the CNN model as a classification task. A triplet loss is employed to optimize the similarity of visualizations in the same type, and the dissimilarity of visualizations in different types. After that, we can feasibly classify a visualization image and represent the result as an 11-dimensional vector, with each dimensional value representing the probability of being the corresponding visualization type. We apply the trained model to several visualization collections, and show the difference among them (Section 6.2).

Lastly, in the *visualization* stage, the embedding vectors are visualized by a CMV system, which consists of 1) *Embedding Overview* that projects the embedding vectors in a radial layout, 2) *Embedding Histogram* that depicts histogram of embedding vectors for each visualization type, and 3) *Visualization Gallery* that shows a gallery of visualizations based on user query. We implement the front-end visualization modules in D3.js, and integrate the modules and visual query using jQuery.

4 Data and Model

We opt for deep learning techniques to meet the requirement of effective image embedding (**R1**). To this end, we first build a new dataset consisting of synthetic and real-world visualizations (Section 4.1), and use it to train a CNN model for neural image embedding (Section 4.2).

4.1 Data Collection

The scope of this work is constrained to static visualization images that are the most common type of data visualization [72]. To improve the effectiveness of image embedding, we enrich the dataset coverage with synthetic and real-world visualizations.

Synthetic Visualization. Synthetic visualizations have been demonstrated effective for learning-based visualization design, e.g., [39], [72]. We develop an automatic method to synthesize visualization images, by carefully considering the rich diversity of data visualization in the following perspectives.

- *Chart types.* As described in Section 3.1, this work considers visualizations of 11 chart types, i.e., *Area, Bar, Circle, Diagram, Grid & Matrix, Line, Map, Point, Table, Word,* and *Trees & Networks*. *Word* and *Table* types are rendered directly in HTML5, whilst the other chart types are all rendered using D3.js. We carefully consider different parameter settings for each chart type, such as the width of bars and space between bars in *Bar* charts.
- *Data distribution.* We employ Rdatasets[1] that contains over 1,300 actual datasets, and synthesize one- and two-dimensional data in normal, beta, and Poisson distributions using SciPy. Specifically, for *Map* type, we utilize the terrain GeoJSON data from GADM[2] that contains 115 administrative countries and regions at all levels of division. For *Word* type, the input data is randomly crawled text from the Guardian News. For flowcharts in *Diagram* type and the graphs in *Trees & Networks* type, inspired by the synthetic methods in [17], [61], we use python to randomly generate node and link data.
- *Colormaps.* We employ frequently-used colormaps from ColorBrewer [1], colorcet [2], and D3 [3]. We use all continuous colormaps, and part of discrete colormaps with 3∼10 colors that are sufficient for most visualizations. Duplicate colormaps are filtered, yielding a total of 256 colormaps in the end.

For each rational combination of these parameters, we render a visualization in a frontend web server and save it to a backend server. All synthetic visualizations are of the size 512×256. For each chart type, we synthesize 1,200 visualizations, yielding a total of 13,200 images for all chart types.

Real-World Visualization. To further enrich the diversity of the visualization collection, we crawl real-world visualization images from the web using Google Images search engine. The engine covers many sources such as medias, research papers, and government reports. We use the visualization subtypes listed in [8] as keywords for each primary type, such as *area chart, proportional area chart* for *Area* type, and *flow map, geographic map, statistical map* for *Map* type. The first 200 images in different sizes are kept for each keyword search. We then manually filter out images in the following cases: i) duplicates in each visualization type; and ii) with wrong labels. In this way, we collect about 400 500 real-world visualizations for each visualization type including *Diagram*, and a total of 4,642 visualizations for all types.

There are in total 17,842 synthetic and real-world visualization images in the collection, which are available in https://osf.io/fnu59/. Each visualization image is assigned to its primary visualization type. The visualizations are used to train a CNN-based classification model for neural image embedding, as described below.

4.2 Image Embedding via CNN

We employ CNNs that have been successfully used in computer vision and image processing, to transform visualization images into human interpretable embeddings. CNN processes an image through multiple layers of convolution operation and non-linear activation to find patterns that are valuable for the downstream tasks of image comparison and query. The model is trained on the visualization collection with synthetic and real-world visualization images, denoted as $\{I_n\}_{n=1}^{N}$ where N indicates the total number of training images. Each image I_n is labeled with an 11-dimensional vector p representing the probability distribution of the visualization in each visualization type. As shown in Fig. 2, the neural image embedding model is built in the following way.

Network Architecture. We adopt ResNet50 [29] as the backbone structure. The architecture consists of multiple blocks of convolution layers with residual connections, which harnesses the power of deeper layers for more complex feature extraction. In particular, the model has 5 convolution stages. The first stage contains a convolution layer using a 7×7 convolution kernel, and a max-pooling layer that takes the maximum value from a

1. https://vincentarelbundock.github.io/Rdatasets/
2. https://gadm.org/

orized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 28,2025 at 20:09:02 UTC from IEEE Xplore. Restrictions a

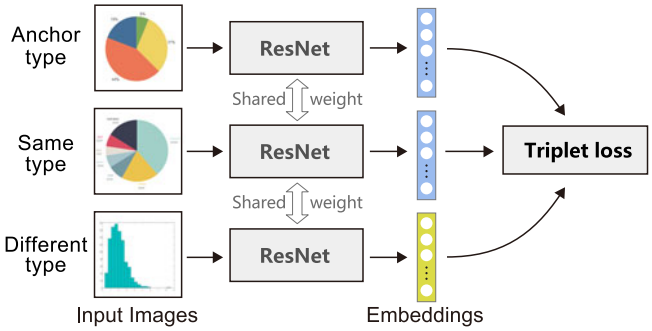


Fig. 2. This work adopts ResNet50 architecture for the neural image embedding model. The model is trained on a classification task with a triplet loss to refine embedding results.

3×3 kernel. The rest of the stages consist of bottleneck convolution blocks with a 3×3 convolution layer in the middle and a 1×1 convolution layer at each end. The residual connection is built between the input and output of each block. The output of the last convolution layer is transformed to a 1000 dimensional vector through a global average pooling layer, which is more effective than directly flattening the tensor. Finally, we add one dense layer with softmax activation to map the predictions to an 11-dimensional vector normalized to [0, 1]. Each value represents the predicted probability for a certain visualization type in our taxonomy.

Loss Function. To supervise the network training, we first employ a *cross entropy loss* to measure the distance between the ground-truth p and the predicted probability distribution q,

$$L_{CE} = \frac{1}{N}\sum_{n=1}^{N} H(p_n, q_n) = \frac{1}{N}\sum_{n=1}^{N}[-\sum_{x=1}^{11} p(x)\log q(x)]. \quad (1)$$

To improve the accordance between embedding values and the similarity of visualizations (**R1.2**), we further introduce a *triplet loss* that is widely adopted in deep metric learning to enhance the representation for similarity measurement [55], as

$$L_{TR} = \frac{1}{N}\sum_{n=1}^{N} max(0, d(q_n, q_m) - d(q_n, q_l) + \alpha), \quad (2)$$

where q_n, q_m, q_l are predicted probability distributions for input images I_n, I_m, I_l respectively, and d is the distance function. Specifically, I_n is the anchor image and I_m belong to the same visualization type of I_n, whilst I_l is a different type. A distance threshold α empirically set to 0.2 is applied to separate similar images and dissimilar images. The triplet loss explicitly drives embedding values of the same type closer to each other than those of different types.

The loss function is set to be the weighted sum of the cross entropy loss and the triplet loss as $L = L_{CE} + \beta L_{TR}$, where the weight β is empirically set to 0.5.

Training Details. Following conventions in computer vision, we randomly select 85% (i.e., $N = 15,165$) images from the visualization collection for training, 5% (893) for validation, and the remaining 10% (1,784) for testing. The network is implemented using Tensorflow and trained on a workstation with two NVIDIA GeForce RTX 3090 GPUs. We use the Adam optimizer with the learning rate set to 0.00005, and the initial model weights pretrained on ImageNet. The network converges after 2-3 epochs and we stop the training.

5 VISATLAS INTERFACE

We design VISAtlas interface that fulfills the requirements of intuitive visual representation (**R2**) and efficient user interaction (**R3**). As illustrated in Fig. 3, VISAtlas interface mainly consists of: a) *Embedding Overview* (Section 5.1), b) *Embedding Histogram* (Section 5.2), and (c)

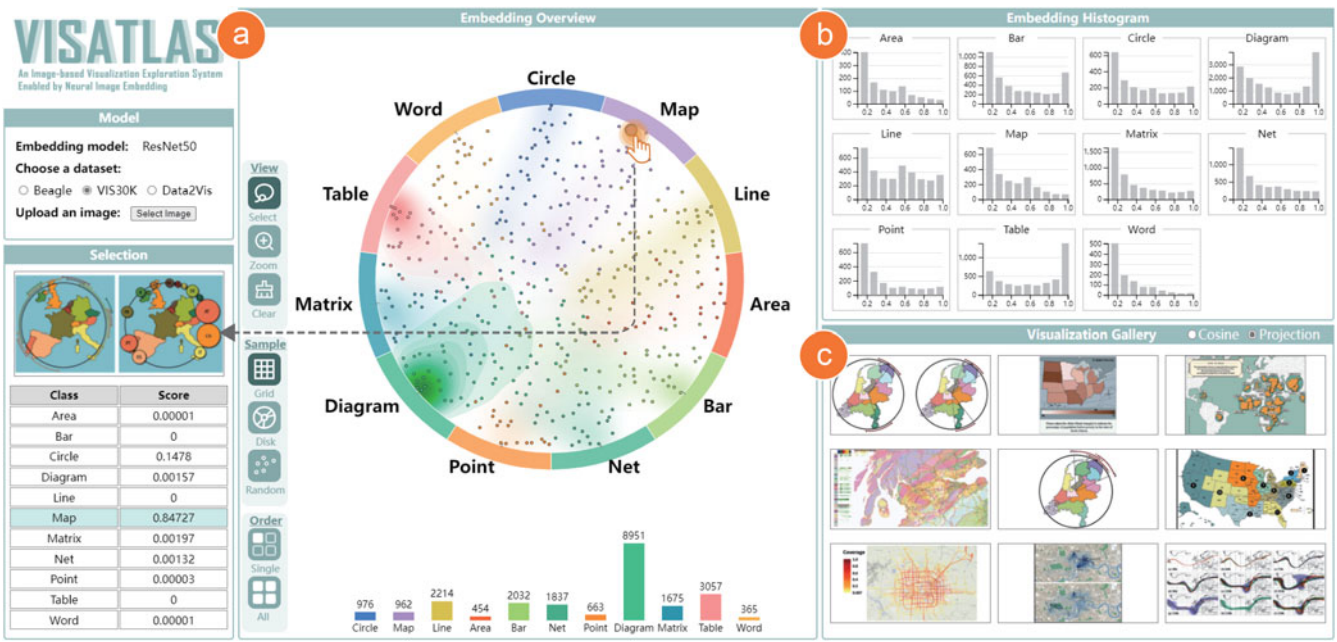


Fig. 3. VISAtlas interface mainly consists of (a) *Embedding Overview* that projects embedding vectors of a selected visualization collection, (b) *Embedding Histogram* that depicts embedding histogram of each visualization type, and (c) *Visualization Gallery* that shows a gallery of visualizations similar to the user-input one. The views are coordinated and complement each other to enable multi-perspective exploration.

orized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 28,2025 at 20:09:02 UTC from IEEE Xplore. Restrictions a

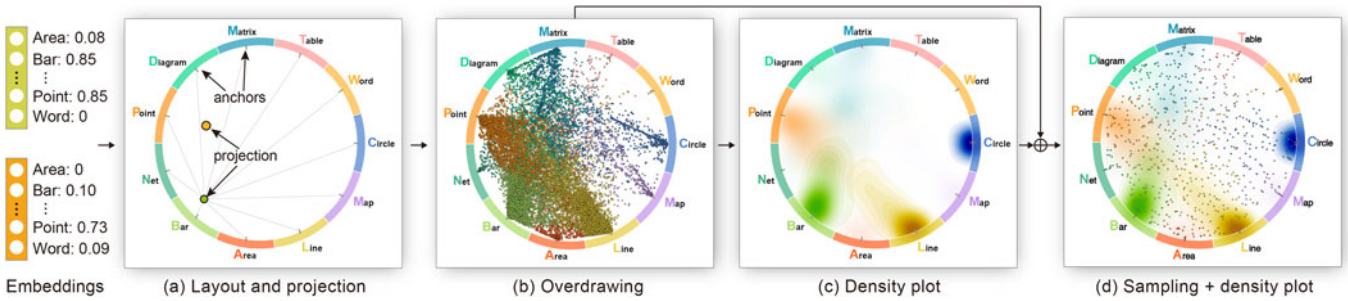


Fig. 4. *Embedding Overview* construction process. (a) Input embedding vectors are first projected onto a 2D plane in radial layout with uniformly placed visualization types. (b) However, projection of all embedding vectors will cause overdrawing problem with points occluding each other. (c) Density plot depicts the estimated density of data point distribution. (d) Combination of sampling and density plot produces the result.

Visualization Gallery (Section 5.3). User interactions including filtering and visual query are enabled to facilitate user exploration and query (Section 5.4).

5.1 Embedding Overview

We design *Embedding Overview* (Fig. 3a) to overview the embedding vectors of a selected visualization collection. As described in Section 3.1, we choose to encode visualization images as human interpretable embeddings generated by taxonomical classification. To preserve interpretability, we would like to simultaneously present 1) the distance of embedding vectors, and 2) embedding vectors in respect to each visualization type (**R2.1**). We opt for contextual layout framework [18] that is able to preserve the context of the data (i.e., embedding vectors) with the associated attributes (i.e., visualization types). We choose to project the embedding vectors onto a 2D plane in radial layout, similar to RadViz [30]. We further employ sampling and density plot techniques [46] to address overdrawing problem when visualizing a large-scale visualization collection.

As illustrated in Fig. 4, *Embedding Overview* is constructed through the following procedures.

1) *Attribute layout*. The visualization types are uniformly arranged on the circumference of a circle with radius r, as shown in Fig. 4a. The center of each type's arc serves as the anchor point for determining the locations of the embedding vectors. Each visualization type is assigned a unique color hue. The visualization types are arranged in an order such that types with more visual similarity are closer to each other as in the training dataset. For example, *Table* type is adjacent to *Word* type because both of them contain texts.

   Alternatively, users can choose an optimal ordering of the attributes according to the visual quality of projection. For this purpose, we use the Davies-Bouldin index [22] to measure the quality of projection in terms of cluster separation [12]. For example, for Beagle dataset, the ordering with optimal visual quality is *Circle, Line, Diagram, Tree & Network, Area, Map, Table, Word, Bar, Point, Grid & Matrix*, with the corresponding Davies-Bouldin index value 0.8065. The optimal ordering for each dataset is precomputed offline. Users can also arrange the dimensions based on their own preference by dragging and dropping attribute arcs in the *Embedding Overview* or bars in the *Type Histogram* below.

2) *Embedding projection*. Location of an embedding vector is then determined by a weighting formula where visualization type with a higher value receives a higher weight that increases the attraction of the location to the corresponding anchor point. Given embedding vector q_n for a visualization image I_n, a spring is set between its projected position P_n and each anchor point, which applies a pulling force to P_n. The stiffness of each spring is set to be equal to the value of q_n in this dimension. The forces from all the springs are then calculated according to Hooke's law. The final P_n is where the forces are balanced as defined by the mapping function:

$$P_n = \sum_{i=1}^{11} \left(\frac{q_n^i}{\sum_{k=1}^{11} q_n^k} v_i \right), \tag{3}$$

   where v_i denotes the vertice's function as $v_i = (r \cdot cos\,\frac{2\pi i}{11}, r \cdot sin\,\frac{2\pi i}{11})$, and q_n^i represents the value in the i-th dimension in the embedding vector q_n. The projection mechanism ensures that embedding vectors with high values in certain visualization types will be positioned closer to the corresponding anchor points. Each projection point is colored according to its primary visualization type, i.e., the type with the highest value in an embedding vector. As such, the green embedding vector is positioned nearby the anchor point of *Bar*, while the yellow one is positioned close to *Point*.

   However, for large datasets, plotting all the embedding vectors would cause the overdrawing problem as illustrated in Fig. 4b. The problem is particularly serious in dense regions of the plot, where points occlude each other. The occlusion leads to less effective overview as the distribution cannot be perceived clearly, and the occlusion hinders user exploration of specific data points. To improve the scalability (**R2.2**), we combine density plot with sampled points to address the overdrawing problem.

3) *Density plot*. The density plot is computed from projections of all embedding vectors using the Gaussian kernel density estimation method. Here we use the same color hue as the visualization type, with sequential lightness to indicate density value. In this way, the density plot presents context information of

orized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 28,2025 at 20:09:02 UTC from IEEE Xplore. Restrictions a

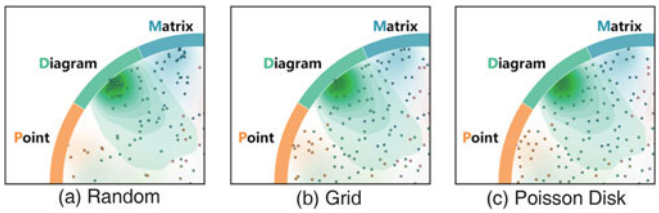


Fig. 5. Comparison of different sampling methods: (a) *Random sampling* produces samples more consistent with the density distribution of the whole collection but causes overdrawing in dense area and inadequate sampling of sparse area; (b) *Grid sampling* shows closer neighboring points in dense area but also keeps points in sparse area; (c) *Poisson disk sampling* produces more evenly distributed samples.

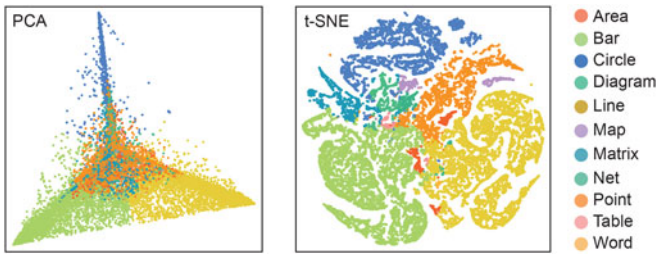


Fig. 6. Alternative visualizations generated by PCA and t-SNE.

the overall embedding distribution. Fig. 4c is the density plot computed from Fig. 4b. Obviously, *Bar*, *Line*, *Circle*, and *Point* have higher density values than other visualization types, and data points around *Circle* are more concentrated.

4) *Sampling + density plot*. Last, we combine the density plot with sampling of the overdrawing projection. The resulting visualization is illustrated in Fig. 4d. Here we provide different sampling methods including *random sampling*, *grid sampling*, and *Poisson disk sampling*. Fig. 5 illustrates a comparison of the results by the sampling methods. Specifically, *random sampling* randomly sample a given number of data points from the original collection. The result (Fig. 5a) better accords with the density distribution, such as more points by *Diagram* anchor point. However, the sampling method may generate overdrawing samples in dense areas but few samples in sparse areas. *Grid sampling* overcomes the issue by dividing the plot into equal-sized grids and sampling at most n points in each grid. In this work, we set the grid size as 30×30 pixel, and n as 2. The result (Fig. 5b) shows closer neighboring points in dense areas, while still keeps samples in sparser areas. Lastly, *Poisson disk sampling* starts from a random point and iteratively adds a randomly sampled point satisfying the distance constraint that sets a lower bound r for the distance between any two sampled points. The default value for r is set to 10 pixels. The result (Fig. 5c) presents more evenly distributed samples in the space. In summary, all the sampling methods make certain tradeoff between accordance with the density and overdrawing issue. Users can switch between the sampling methods on an as-needed basis.

Alternative Design. Fig. 6presents two alternative designs for *Embedding Overview*, using dimension reduction methods of PCA and t-SNE. The points are colored according to the primary visualization types, and the points are randomly ordered when rendering. The input embedding vectors are the same as those used in Fig. 4. From the figure, PCA projection can better preserve the global structure in the four most prominent visualization types, i.e., *Bar*, *Line*, *Circle*, and *Point*. However, embedding vectors of visualization in less prominent visualization types tend to be overshadowed by these dominant ones. In comparison, t-SNE projection depicts the local clusters more clearly, where more visualization types are appearing. On the other hand, t-SNE projection trades off the ability in retaining distances of embedding vectors in different visualization types, and the result is very sensitive to its perplexity parameter.

Alternatively, we can employ other projection methods like MDS [21] or UMAP [47]. However, a common problem is that the methods are not able to depict embedding values in relation to each visualization type (against **R2.1**), making the visualizations less interpretable. Another problem is the varying output space due to the optimization of the variance (PCA) or the distance (t-SNE). When the methods are used on different visualization collections, direct comparison of the resulting plots is less reliable because different collections will go through different transformations. Instead, our method has a fixed transformation scheme that will preserve the structure.

Type Histogram. The density plot depicts the distribution of different visualization types, but not the real numbers. To compensate this, we turn the legend of *Embedding Overview* into a scented widget design [67] of histogram below the radial plot. The number of data points for each visualization type is presented on top of the bar chart. As shown in Fig. 3a, the histogram depicts that *Diagram* is the most frequent visualization type, which is almost three times than the second type of *Table*.

5.2 Embedding Histogram

We design *Embedding Histogram* (Fig. 3b) that provides a more detailed channel to explore the embedding vectors (**R2.3**). For each visualization type, we collect all embedding vectors that are higher than 0.1 in this dimension. The other vectors are omitted because they are of little interest. Next, we discretize the range of [0.1,1] into nine equal intervals, and count the number of embedding vectors within each range. We dynamically set the range of y-axis to fit the number of visualizations in each histogram. Last, we order the histograms according to the type name, and organize them as small multiples.

The histograms show interesting patterns among distributions for each type. For instance, among the four most common visualization types in Fig. 3, *Line*, *Circle*, and *Bar* types have dominant distributions in [0.9,1] interval, whilst *Point* type is more evenly distributed. We select small-value intervals in *Point* histogram, and notice that many of these visualizations are actually *Trees & Networks* type. It is challenging for the embedding model to distinguish graphs with thin edges from *Point* visualizations.

5.3 Visualization Gallery

We develop *Visualization Gallery* (Fig. 3c) to show query results of similar visualizations in the collection with user-

orized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 28,2025 at 20:09:02 UTC from IEEE Xplore. Restrictions a

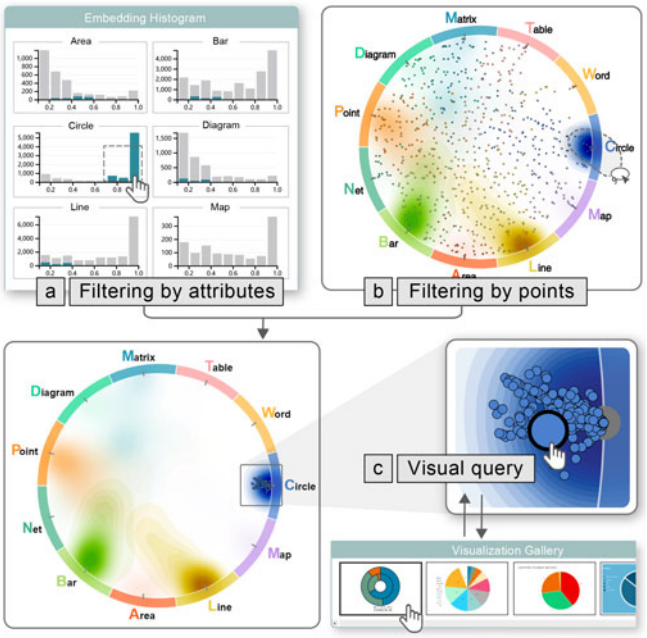


Fig. 7. VISAtlas enables i) *filtering* by attributes or data points, and ii) *visual query* by data point selection or image uploading, to support efficient user interaction.

input visualization by selecting a data point in *Embedding Overview* or uploading an image (**R3.2**). Detailed information of the user-input visualization, including the image itself and its embedding vector, are presented in *Selection* panel on the left. The system computes an embedding vector for the selected/uploaded visualization using the CNN-based image embedding model. The vector is used as a query input to traverse the whole dataset and find 9 visualizations with the smallest distances. Here, we offer two options to measure distance for the query: i) cosine similarity between embedding vectors, and ii) euclidean distance between projected points in *Embedding Overview*. We use linear search to find the most similar visualizations, which is fast enough for real-time performance tested on the three visualization collections. The top 10 visualizations are displayed side by side in *Visualization Gallery*. Users can select one from the query results, which will update the input visualization and repeat the query process.

5.4 User Interaction

To support efficient exploration and query, VISAtlas mainly provides the following two categories of user interactions:

- *Filtering*. Users can filter data items by attributes or by data points. For filtering by attributes, users can select a range of bins in histogram of a specific visualization type in *Embedding Histogram* (Fig. 7a). Alternatively, user can directly select data points using lasso selection tool enabled in *Embedding Overview* (Fig. 7b). Upon filtering, *Embedding Overview* will display the filtered data points in full details without sampling, and keep the initial density plot to preserve context information. Histograms in *Embedding Histogram* will show distributions of the filtered data, and *Type Histogram* will also adjust the bar lengths to depict the numbers of visualization types in the filtered data. In this way, linked brushing is enabled to promote multi-perspective exploration (**R3.1**).
- *Visual query*. Users can query similar visualization designs (**R3.2**) by selecting a data point in *Embedding Overview* (Fig. 7c), or uploading a visualization image through the control panel. *Selection* panel (Fig. 3) on the left will display the image and the embedding vector. The selected/uploaded visualization will be highlighted as a circle with a larger radius and thicker ring. The operation will trigger the query function and the query results are displayed in *Visualization Gallery*, as described in Section 5.3.
- *Zooming*. The sampling + density plot approach can scale up to large visualization collections, yet users may find it difficult to inspect local details. Therefore, we implement the zooming interaction as a complement. Users can inspect local points with a zooming lens, with mouse wheel used to adjust the zooming level. Users can also fix the zooming lens by pressing the mouse's middle button and inspect details of the points shown within the magnification window.

6 Evaluation

This section evaluates the performance of our CNN-based embedding model (Section 6.1), followed by three case studies on the applicability of VISAtlas in comparing visualization collections (Section 6.2), exploring composite visualizations (Section 6.3), and retrieving visualization designs (Section 6.4). In the end, we present user feedback of *VISAtlas* system (Section 6.5).

6.1 Model Performance

We evaluate classification performance of the CNN-based embedding model. To validate the necessity of real-world visualizations, we first compare two backbone embedding models without triplet loss: one trained on the synthetic visualizations alone versus another one trained on both synthetic and real-world visualizations. For both models, we randomly sample 85%, 5%, and 10% from each type of the datasets for training, validation, and testing, respectively.

Table 1 shows the top-1 classification performance in terms of precision, recall, and F1-score of the two models for each visualization type. The overall F1-score for the backbone model trained on both synthetic and real-world visualizations is 94.36%, whilst that for the one trained on synthetic visualizations is 82.26%. In general, combining synthetic and crawled data for training can significantly improve model performance across different types. More specifically, Table 1 shows that the performances for several visualization types such as *Diagram* and *Map* have been boosted the most with real-world visualizations. A possible reason is that real-world visualizations provide more complex visual features for these visualization types. For example, D3-generated maps usually have no texture, while real-world maps typically have complex terrain textures. On the other hand, there is less improvement for simpler chart types like *Area*, *Bar* and *Line*, probably because of their

orized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 28,2025 at 20:09:02 UTC from IEEE Xplore. Restrictions a

TABLE 1
Test Classification Performance of the Backbone Models (Without Triplet Loss) Trained on the Synthetic Visualizations Alone versus on Both Synthetic and Real-World Visualizations

	Synthetic			*Synthetic+real*		
	Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*
Area	96.27%	91.89%	93.43%	**97.31%**	**95.29%**	**96.29%**
Bar	93.33%	83.80%	88.31%	**96.16%**	**88.38%**	**92.11%**
Circle	85.92%	87.19%	86.55%	**99.50%**	**98.02%**	**98.75%**
Line	**97.01%**	78.31%	86.66%	95.73%	**94.57%**	**95.15%**
Matrix	82.00%	68.47%	77.77%	**84.57%**	**92.39%**	**88.31%**
Map	67.28%	93.16%	78.13%	**97.19%**	**94.87%**	**96.31%**
Point	98.05%	68.47%	80.79%	**99.44%**	**86.47%**	**92.50%**
Table	74.92%	**96.12%**	82.32%	**95.34%**	96.09%	**95.71%**
Text	78.87%	90.32%	84.21%	**98.37%**	**97.58%**	**97.97%**
Net.	89.47%	72.34%	80.00%	**90.40%**	**95.21%**	**92.74%**
Diagram	87.68%	45.66%	60.04%	**87.85%**	**82.26%**	**85.49%**
Overall	85.37%	81.86%	82.26%	**95.24%**	**93.49%**	**94.36%**

Bold *numbers indicate the highest metric value for each type. The overall metric values in the last row are weighted average based on number of each type in test set.*

simpler visual features and the smaller gap between the synthetic and real world images.

There are also some counterexample including *Line* type with a dropping precision, *Table* type with a dropping recall. For *Line*, we checked the results and found many real-world *Network* visualizations in horizontal layout are misclassfied as *Line* type. For *Table*, we found that the recall is higher for the synthetic model because the model cannot distinguish *Table* well from other visually similar types such as *Matrix* and overconfidently classify more images as *Table*. Moreover, Table 1 also reveals that the classification performance for different visualization types are diverse, even for the model trained with real-world visualizations. The model achieves relatively low performance for *Matrix* and *Diagram* types. For *Matrix* type, a possible reason is that the model misclassifies between *Matrix* and *Table* visualizations (see Section 6.3 for more discussions). For *Diagram* type, this is because *Diagram* includes various complex sub-categories such as flow charts and scientific visualization diagrams which have complex features that are more likely to be confused with other types. For example, flowcharts with small nodes and thin arrows may be confused with *Network*.

We further evaluated the choice of the neural network approach by comparing with two traditional machine learning methods (SVM [9] and Random Forest [11]), the choice of using triplet loss by comparing with with model without triplet loss, and *ResNet* as the backbone architecture by comparing with other two commonly used CNN backbones (VGG19 [60] and InceptionV3 [62] with triplet loss). Table 2 shows the precision, recall, and F1 scores. All models are tested on the testing dataset with both synthetic and real-world visualizations. The results show that CNN-based models are better than traditional machine learning models, while there is no significant difference between the backbone architectures with F1 scores above 95%. Also, triplet loss improves the classification performance with about 2% increase of F1 score. Nevertheless, ResNet achieves slightly better performance than the others. As such, we choose ResNet as the backbone architecture for the embedding model. More evaluation results to compare the ResNet model trained on different collections can be found in Supplementary Table S1.

TABLE 2
Comparison of Different Models and Backbone Architectures

	Traditional		Neural			
	SVM	RF	ResNet	ResNet+	Inception+	VGG+
Precision	87.23%	84.70%	95.24%	**96.38%**	95.49%	95.36%
Recall	85.55%	84.05%	93.49%	**96.29%**	95.46%	95.10%
F1	86.74%	84.52%	94.36%	**96.30%**	95.48%	95.17%

+ symbol after model name indicates the model is trained with triplet loss.

6.2 Study 1: Comparing Visualization Collections

VISAtlas can be utilized to examine and compare the richness and diversity of visualization images in different collections, to help machine learning developers for visualization choose appropriate datasets. In this study, we compare three visualization collections, including Beagle [5], VIS30K [14], and Data2Vis [25]. Fig. 8 presents Embedding Overviews for the collections. The visualizations reveal significant differences, as follows:

- *Beagle*. The Beagle dataset is a collection of over 41,000 visualization images crawled from the web. As in Fig. 8a, Beagle collection shows higher densities around *Line*, *Bar*, *Circle*, and *Point*. A close examination reveals that the collection contains many simple bar charts, line charts, pie charts, and scatter plots, which are generated by web-based visualization tools like D3, Plotly, and Chartblocks. In comparison, there are seldom complex visualization types like *Table*, *Word*, *Trees & Networks*, and *Diagrams*.
- *VIS30K*. The VIS30K collection contains over 29,000 images extracted from the papers published in the IEEE Visualization conferences between 1990 and 2019. The embedding overview (Fig. 8b) shows a more balanced distribution among the visualization types. An exception is *Diagram* type that has the highest density and the most number. We find that there are many flow charts, illustrations, and scientific visualizations, which all belong to *Diagram* type. The finding is consistent with the source of VIS30K which includes IEEE SciVis for scientific visualizations. Moreover, academic papers often use flow charts to illustrate the methodology or demonstrate the experiments.
- *Data2Vis*. The Data2Vis is a collection of over 4,300 visualizations synthesized with Vega-Lite. The major difference between the synthetic dataset and the other two real-world datasets is the limited diversity and richness of visualization types. Fig. 8c and shows that visualizations in Data2Vis are concentrated in four types: *Point*, *Line*, *Area*, and *Bar*. Compared to the other two collections, Data2Vis is less representative of the diverse design space of visualization. This reflects a common problem of current researches on data-driven visualization design: for

orized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 28,2025 at 20:09:02 UTC from IEEE Xplore. Restrictions a

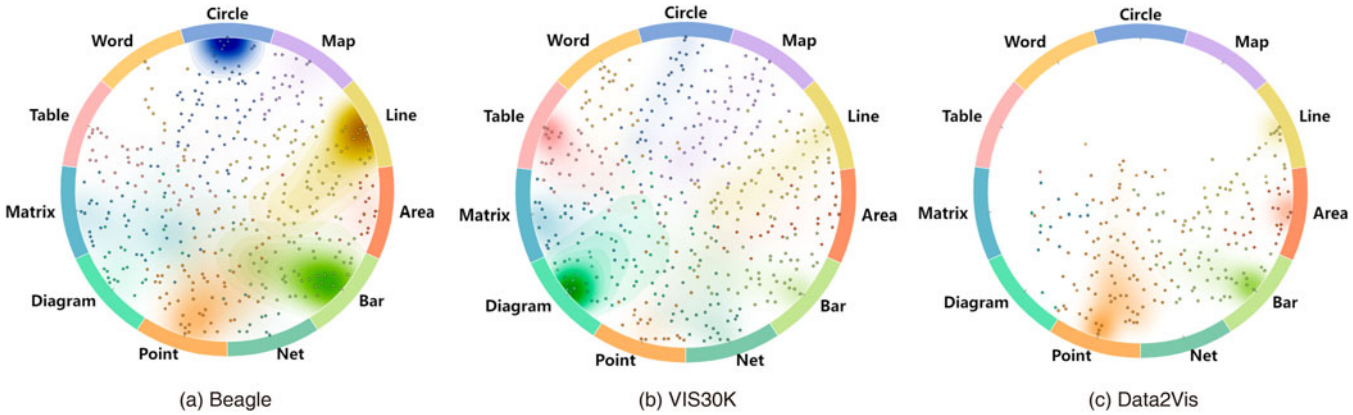


Fig. 8. Comparison of visualization collections: (a) Beagle of visualizations from the web[5], (b) VIS30K of visualizations from IEEE VIS publications [14], and (c) Data2Vis of synthetic visualizations[25]. Real-world datasets (Beagle and VIS30K) better reflect the richness and diversity of visualization design than the synthetic one (Data2Vis).

the sake of model simplicity, existing works only focus on a limited range of visualization types.

The study supports the necessity for collecting real-world visualizations. We are delighted to see more efforts devoted in this direction recently, such as [8], [14], [16], [24], [54]. This work complements the visualization collections with VISAtlas that helps compare and identify limitations in the collections. For example, Beagle does not provide enough *Table*, *Word*, *Trees & Networks*, and *Diagrams* visualizations. VIS30K would be a better choice if someone is interested in developing automatic methods for flow charts or scientific visualizations. More analysis results for other collections can be found in Supplementary Fig. S1 and Fig. S2.

6.3 Study 2: Exploring Composite Visualizations

Composite visualizations combine different visual representations in the same geometric space to take advantages of strengths and overcome weakness of individual visualizations [34]. As such, composite visualizations are typically recognized as embeddings without a dominating dimension by our model. VISAtlas can help users explore composite visualizations in a visualization collection, using the filtering function enabled in the *Embedding Overview* and the *Embedding Histogram*. In the following, we use examples found in VIS30K dataset, showing how VISAtlas can be utilized to identify composite visualizations and explore the interplay between different visualization types.

As shown in Fig. 9a, the user first performs a histogram filtering by selecting bins of range [0.3, 0.6] in *Matrix* dimension. The model may classify the selected points to be *Matrix*, but without high confidence. The points are displayed in *Embedding Overview*, and the user further selects data points that are not close to any anchor point (Fig. 9b). The selected region has some distance from the anchor point of *Table*, yet there are still many points with embedding values in the range [0.2, 0.6] in *Table* dimension (Fig. 9c). The user performs a second histogram filtering in the range, yielding data points with compatible embedding values in both *Matrix* and *Table* dimensions.

To this end, users can select the data points of interest by highlighting and inspecting the details in the *Selection* panel. Fig. 9e1 presents an example with almost equal embedding values in *Matrix* and *Table* dimensions. The visualization is indeed a table visualization with heatmaps showing the difference between numbers in the table cells. Similar designs are observed in examples 2–5 in Fig. 9e. By observing these visualizations, designers can get some inspiration of what visualizations can be composited, to enhance the intuitiveness and information presented in a single visualization.

6.4 Study 3: Retrieving Visualization Designs

Supporting image-based query of visualization design is another main goal of VISAtlas system. This could help visualization novices to explore similar images and learn from the rich design patterns in visualization collections. In this study, we demonstrate how VISAtlas can help retrieve visualization designs through visual query. Fig. 10 shows some example results for both real and hand-drawn visualizations of different chart types from two real-world datasets.

- *Mark types*. VISAtlas supports effective visual query for a variety of chart types with different visual marks. In Fig. 10, visual marks of heatmap (row 1), area chart (row 2), and pie chart (row 4) are area type, graph visualization (row 3) is both point and line type, bar chart (row 5) is line type, and scatterplot

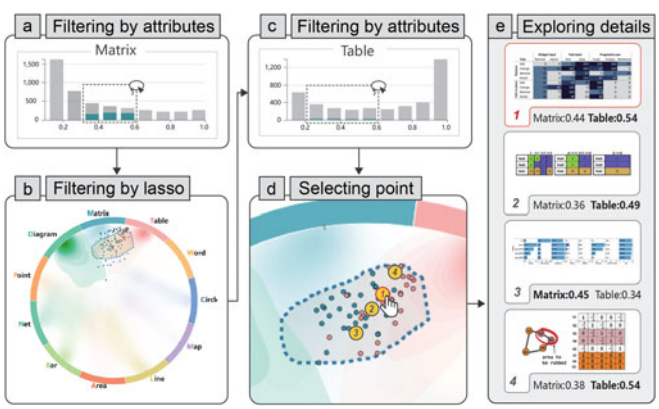


Fig. 9. Exploring composite visualizations: (a) Filtering by attributes by selecting bins in *Matrix* dimension of *Embedding Histogram*, (b) filtering by lasso region in *Embedding Overview*, (c) a second histogram filtering in *Table* dimension of *Embedding Histogram*, and (d) selecting to explore details of the composite visualizations.

orized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 28,2025 at 20:09:02 UTC from IEEE Xplore. Restrictions a

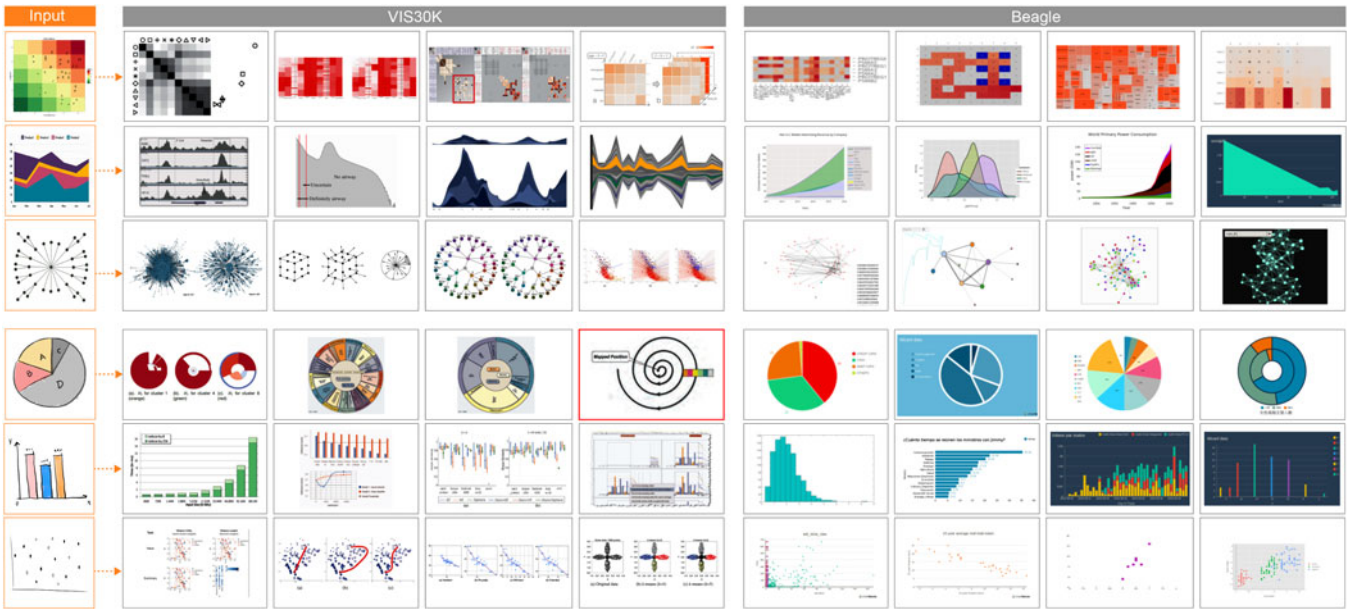


Fig. 10. VISAtlas allows users to retrieve visualization designs for both real and hand-drawn visualizations of different chart types from different datasets. Users can find similar visualizations that can be used as design references.

(row 6) is point type. All query results are in the same type with the input visualization, except that the spiral line graph highlighted in red (row 4 column 4) is misclassified as a pie chart. The query is robust to diverse visual channels including different colormaps (e.g., grayscale versus chromatic) and layout arrangements (e.g., horizontal versus vertical). The results demonstrate the effectiveness and robustness of the embedding model in handling different visual marks and channels.

- *Real versus. hand-drawn inputs*. VISAtlas supports both real visualizations and hand-drawn sketches as inputs. In Fig. 10, the inputs in rows 1−3 are real visualizations, while those in rows 4−6 are hand-drawn sketches. Supporting hand-drawn sketches is of importance for novices demanding to better understand data visualization, and designers looking for inspiration or checking if similar designs exist in visualization collections. For example, the user uploads a scratch pie chart in row 4. She learns that the center of a pie chart can be utilized to display more information, as shown in the results from VIS30K. Users can also find other design alternatives from the retrieved real-world examples, such as using a two-layer donut chart to compare two distributions, placing the text in different positions, using grid lines for bar charts, and using a single hue color map with different lightness. Notably, we find that the system supports real visualizations better than hand-drawn ones. This is natural since our embedding model is only trained on synthetic and real-world visualization, whilst hand-drawn sketches have unique visual features like wiggly lines. This may partially explain why the input of a hand-drawn pie chart (row 4) returns a spiral line graph highlighted in red.
- *Different datasets*. The user can compare design patterns through results retrieved from different visualization collections. In Fig. 10, the results in columns 1−4 are from VIS30K, while those in columns 5−8 are from Beagle. Here, an obvious difference is that query results in Beagle tend to use more dark background, whilst those in VIS30K use simple white background. The difference reminds users that are making images for academic papers to adopt white background for clarity. Another interesting difference is that VIS30K contains more visualizations with side-by-side charts, whilst the results in Beagle seldomly show multiple charts in one image. This is probably because academic papers frequently use juxtapositioned charts for results illustration and comparison. From a new aspect, the result confirms the effectiveness of juxtaposition for comparison [26].

6.5 User Study

The utility, usability, workload and visual design of *VISAtlas* are further evaluated in a user study.

Participants: We recruited 12 participants (5 males and 7 females; age: 21-27, average 24) for the study. All the participants are postgraduate students from different backgrounds including computer science, industrial design, and civil engineering. They all have a basic knowledge of data visualization gained from the visualization course. Specifically, 7 out of 12 have experience in developing a visualization system. After the study, each participant was compensated with a gift of about $20 after the study.

Procedure: The study was conducted in a one-on-one manner remotely due to the pandemic. In each study, we first introduced the visualization design and system workflow to the participants. Next, the participants were allowed to freely explore the system for 10 minutes. After the participants felt familiar and confident with the system, they were asked to complete three tasks corresponding to the case studies. Task 1 required the participants to compare different visualization collections through *Embedding Overview*, identify 2−4 dominant visualization types in each collection, and find the most

orized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 28,2025 at 20:09:02 UTC from IEEE Xplore. Restrictions a

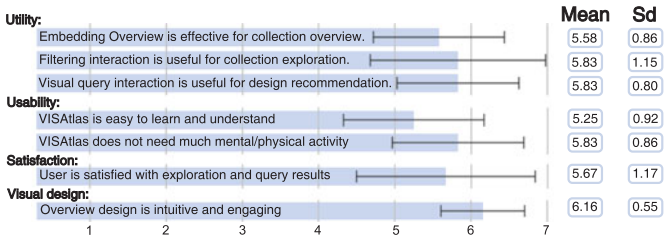


Fig. 11. Summary of survey results. Most participants are in favor of the system's utility, usability, satisfaction and visual design.

balanced collection. Task 2 required the participants to first choose four visualization types, and then identify three typical examples using the system. Task 3 required the participants to upload four visualization images and evaluate whether the retrieval results match the input visualization.

After completing the tasks, the participants were asked to answer seven questions with answers in 7-point Likert scale, and to provide general feedback on visual and interaction design of the system. Each study lasted for about 50 minutes. The detailed questions and ratings are presented in Fig. 11. More results are provided in Supplementary Table S2. Overall, all questions receive ratings higher than 5 on average, which show participants' satisfaction on the *VISAtlas* system.

- *Utility*: The participants acknowledged the utility of the system in terms of visualization collection overview (*mean=5.58, sd=0.86*), collection exploration (*mean=5.83, sd=1.15*), and design recommendation (*mean=5.83, sd=0.80*). *Embedding Overview* and *Embedding Histogram* coupled with filtering interactions are *"particularly helpful for exploring and finding the charts I want" (P8)*. The visual query interaction is also important for visualization designers, as *P5* commented: *"As a designer, I can benefit from the retrieved examples as I can see some alternative layouts never seen or cannot recall".P10* also praised that *"the filtering and zooming interactions make a good combination, allowing to explore the data in any region, in any range, and in greater or less detail"*.
- *Usability*: Most participants agreed that the system has good usability and light workload, including ease of learning (*mean=5.25, sd=0.92*) and no need for much physical or mental activity (*mean=5.83, sd=0.86*). *P11* felt the *Embedding Histogram* needs a bit time to comprehend, especially the multi-step filtering operation. Nevertheless, the participant acknowledged that *"the whole system is straightforward and easy-to-use"*.
- *Satisfaction*: The participants were satisfied with the exploration and query results (*mean=5.67, sd=1.17*), as the system allows to *"see the distribution of charts and find the chart I am looking for" (P3)*.
- *Visual design*: The participants appreciated the visual design of the system (*mean=6.16, sd=0.55*), especially the design of *Embedding Overview*. They were not aware of the radial layout for high dimensional data visualization before the study. After we explained the design and showed the comparison to conventional projections (Fig. 6), they agreed that the design can simultaneously present the distance of embedding vectors, and embedding vectors in respect to each visualization type. Besides, *"sampling + density plot looks very comfortable and aesthetically pleasant, meanwhile reduces visual clutter and makes an engaging plot" (P5)*.

General feedback: Overall, the participants appreciated the design of *VISAtlas* system, as *"the system looks beautiful, especially the layout and the color scheme" (P1)*. However, the participants also gave some fruitful suggestions for improvement.

- On the sampling approaches provided in the *Embedding Overview*, participants have different opinions. Notably, all the participants can clearly discern the difference between random sampling and the other two approaches, but they have different preferences. Specifically, 5 out of 12 prefer *random sampling*, as *"the results (by random sampling) make the scatterplot and the density plot more consistent" (P1)*. In contrast, the rest of the participants prefer *Grid sampling* and *Poisson disk sampling* because of more uniform sample distribution in space that achieves *"a good balance between matching with the density plot and displaying more diverse data points" (P5)*.
- Regarding the ordering of the attributes in the radial layout, all the users think that the ordering does not have significant impact on their understanding and exploration, since the density plot can effectively show the distribution. Nevertheless, the ordering can be useful when exploring composite visualizations, by *"putting the two attributes next to each other to see the intersection area" (P8)*.

7 Discussion

VISAtlas is an image-based exploration and query system for visualization collections. The system is empowered by neural image embedding, coupled with new visualization designs to support the analytical goals. The applicability of VISAtlas in supporting the comparison of visualization collections and retrieval of visualization designs is demonstrated in the studies. The combination of advanced machine learning models with interactive visual analytics techniques enables the functionalities that are not supported in attribute-based visualization exploration systems. Nevertheless, this does not imply that an image-based system is superior than an attribute-based one. Instead, image- and attribute-based methods shall be combined to complement each other in supporting complicated user requirements. For example, when interesting designs are identified from the query results as in Fig. 10, users would like to get more information like the publication and the authors. This is only achievable with finely-annotated visualization collections. The sampling method could also be improved by taking into account meta-data in the sampling process. For example, publication date could be considered to obtain samples that are more concentrated or evenly distributed in time. For academic collections, citation information could be used to prioritize samples from papers with higher citations. Therefore, to provide complete functionality for more

orized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 28,2025 at 20:09:02 UTC from IEEE Xplore. Restrictions a

complex tasks, both image- and attribute-based systems are indispensable.

7.1 Potential Usage Scenario

The construction of visualization collections shows obvious benefits in many aspects, such as to understand design patterns of visualizations, and for teaching and communication [5], [14], [16], [24]. VISAtlas further complements the visualization collections with an image-based exploration and query system, which opens up new usage scenarios such as collection comparison as in Study 1 (Section 6.2) and design retrieval as in Study 3 (Section 6.4). On this basis, VISAtlas shows great potential in benefiting the following applications.

- *Machine learning for visualization*. Researchers on machine learning for visualization design and assessment often need a large dataset to train the model. As in Study 1, VISAtlas can help identify a suitable dataset with an intuitive overview of the image collections. The visual query and filtering interactions in VISAtlas also allow users to explore and discover detailed image characteristics in the dataset. After identifying the limitations of a dataset, developers better understand how to enhance the dataset. For example, users can discover from the overview what visualization types are under-represented. They can then enrich and balance the dataset by supplementing more visualizations of the under-represented types. When augmenting supplementary visualizations, the image embedding model can be used to classify and filter out unnecessary types.
- *Visualization reference*. The visualization community has developed many visualization reference systems, such as TreeVis [56] and TimeViz [4], to help better understand data visualization. However, a visualization reference system requires substantial efforts for maintenance. A comprehensive visualization collection coupled with VISAtlas can help reduce the efforts. As in Study 2, with the image-based query function, users can efficiently retrieve visualization designs using real and hand-drawn visualizations, rather than specifying any keyword. Fig. 12 shows another example of retrieval results from VIS30K for two different types of tree visualizations: node-link graph (top) and radially stacked tree (bottom). The results show similar or related visual designs in the dataset. Users can further examine details of the designs to identify potential applications. For example, some of the design choices in the retrieved examples can be adopted by the designer to refine their own design.

Furthermore, VISAtlas provides a general framework that can be easily adapted to image collections in other domains. One possible scenario is the analysis of training dataset for generative adversarial networks (GANs) that have been widely applied in artistic and realistic image generation, such as the generation of virtual human portraits. An obstacle for effective GAN models is the requirement of large amounts of raw images for training. Diversity and quality of the training data will significantly affect the generation performance. In such scenario, the image embedding model can be trained to classify images of different styles and quality. Then, VISAtlas exploration system can be utilized to analyze the training data, to help answer important questions including whether the dataset is diverse enough and whether the dataset contains some undesirable images.

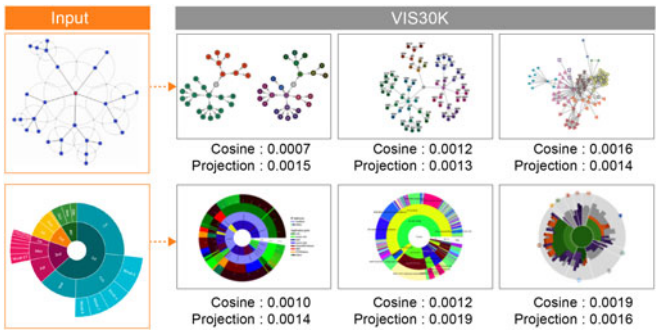


Fig. 12. Query visualizations of a node-link graph (top) and a radially stacked tree (bottom) from VIS30K. The results can help reduce the efforts for maintaining a tree visualization reference system.

7.2 Limitations and Future Work

Nevertheless, there are certain limitations in our work. First, the neural image embedding model is built upon primary visualization types only. We have exploited synthetic + real-world visualization and a triplet loss to improve the prediction accuracy. The experiments have demonstrated the effectiveness and robustness of the model. However, the current model is not enough to reflect the rich diversity of visualization design space, such as diverse colormaps and layout arrangements. As a result, the distance between embedding vectors may not correctly reflect the similarity of visualizations. For better similarity measurement, we plan to improve the model from two aspects: i) From the perspective of data representation, we would like to incorporate fine-grained information of visualizations, including visualization subtypes [8], colormaps [72], text elements, underlying data, and composite patterns [23]. A more fine-grained embedding can also produce better representation for composite visualizations as shown in Study 2 (Section 6.3); ii) From the perspective of embedding model, we can improve model performance using multiple similarity losses at different layers of the CNN model. For machine learning developers for visualization who care more about high model accuracy, they may be interested in actively correcting the misclassified visualizations and fine-tune the model. However, this is beyond the scope of system requirements for this work. In the user study, the participants raised up some concerns regarding the point, since *"user labels may add more bias as different users have different understanding of visualizations. Labels by a single author shall be verified by others" (P9)*. In addition, some participants appreciated that there are some inaccurate visualization retrievals, which help to *"discover unexpected designs from the retrieval results" (P1)*. To address these concerns, we plan to introduce a more reliable active learning mechanism such as crowdsourced labeling followed by fine-tuning on a unified backend server to resolve user differences. Meanwhile,

orized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 28,2025 at 20:09:02 UTC from IEEE Xplore. Restrictions a

we would also like to develop a model that strikes a balance between accuracy and variety to better serve both model developers and visualization designers .

Second, the interactive visualization system also exhibits some limitations. *Embedding Overview* adopts contextual layout framework [18] to present the distance of embedding values, and embedding values in respect to each visualization type simultaneously. However, the radial layout is a non-linear projection that distorts the linear relationship among embedding vectors. As such, the distance between two visualizations is not correctly reflected from the distance between the projected points. We mitigate the issue by enabling query by both cosine (i.e., embedding vector) distance and projection distance. Here, one possible solution is to complement the non-linear overview with another view using linear projection methods such as PCA. Besides, all views in the visualization system are built upon embedding information only, which can be feasibly complemented with other attributes regarding the visualizations. We will build a more complete visual analytics system integrating image- and attribute-based methods.

Last but not least, we would like to increase the generalizability of our system to analyze image collections in other domains. Visual arts is a promising one, due to the rich diversity in design patterns and the demanding requirements for exploration and analysis from art analysts and students. Nevertheless, there may not be enough training data with fine annotations. We plan to exploit visual category discovery methods to group unlabeled images into different semantic partitions [75]. We anticipate to reveal interesting design patterns for visual arts.

8 Conclusion

We have presented VISAtlas − an image-based exploration and query system for visualization collections powered by neural image embedding. The effectiveness of VISAtlas system is boosted from two perspectives: First, we build a comprehensive visualization dataset with synthetic and real-world visualizations, and use it to train a CNN model with triplet loss for taxonomical classification of visualization images. Second, we develop a coordinated multiple view system with novel visualization design to facilitate comparative analysis of visualization collections, exploration of composite visualizations, and image-based visualization retrieval. VISAtlas complements visualization collections and shows great potential in supporting machine learning for visualization and maintaining visualization references. The success of VISAtlas also illustrates the advantage of combining advanced machine learning techniques with interactive visualization systems in addressing complex analytical tasks. We anticipate to apply the framework to other imagery collections to help domain experts better understand their datasets.

Acknowledgments

The authors wish to thank the anonymous reviewers for their valuable comments.

References

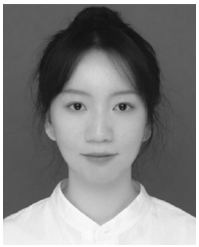
[1] Colorbrewer2, 2013. Accessed: Mar. 01, 2022. [Online]. Available: http://colorbrewer2.org/
[2] Colorcet, 2016. Accessed: Mar. 05, 2022. [Online]. Available: https://github.com/pyviz/colorcet
[3] D3-scale-chromatic, 2018. Accessed: Mar. 01, 2022. [Online]. Available: https://github.com/d3/d3-scale-chromatic
[4] W. Aigner, S. Miksch, H. Schumann, and C. Tominski, *Visualization of Time-Oriented Data*, Berlin, Germany: Springer, 2011.
[5] L. Battle, P. Duan, Z. Miranda, D. Mukusheva, R. Chang, and M. Stonebraker, "Beagle: Automated extraction and interpretation of visualizations from the web," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2018, pp. 1–8.
[6] M. Behrisch et al., "Quality metrics for information visualization," *Comput. Graph. Forum*, vol. 37, no. 3, pp. 625–662, 2018.
[7] M. Berger, K. McDonough, and L. M. Seversky, "cite2vec: Citation-driven document exploration via word embeddings," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 691–700, Jan. 2017.
[8] M. A. Borkin et al., "What makes a visualization memorable?," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2306–2315, Dec. 2013.
[9] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, 1992, pp. 144–152.
[10] M. Bostock, V. Ogievetsky, and J. Heer, "D^3: Data-driven documents," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011.
[11] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[12] L. D. Caro, V. Frias-Martinez, and E. Frias-Martinez, "Analyzing the role of dimension arrangement for data visualization in radviz," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, 2010, pp. 125–132.
[13] P. Chagas et al., "Evaluation of convolutional neural network architectures for chart image classification," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2018, pp. 1–8.
[14] J. Chen et al., "VIS30K: A collection of figures and tables from IEEE visualization conference publications," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 9, pp. 3826–3833, Sep. 2021.
[15] M. Chen et al., "Data, information, and knowledge in visualization," *IEEE Comput. Graph. Appl.*, vol. 29, no. 1, pp. 12–19, Jan./Feb. 2009.
[16] X. Chen, W. Zeng, Y. Lin, H. M. Al-maneea, J. Roberts, and R. Chang, "Composition and configuration patterns in multiple-view visualizations," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 1514–1524, Feb. 2021.
[17] Z. Chen, Y. Wang, Q. Wang, Y. Wang, and H. Qu, "Towards automated infographic design: Deep learning-based auto-extraction of extensible timeline," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 1, pp. 917–926, Jan. 2020.
[18] S. Cheng and K. Mueller, "Improving the fidelity of contextual data layouts using a generalized barycentric coordinates framework," in *Proc. IEEE Pacific Vis. Symp.*, 2015, pp. 295–302.
[19] S. Cheng, W. Xu, and K. Mueller, "RadViz deluxe: An attribute-aware display for multivariate data," *Processes*, vol. 5, no. 4, 2017, Art. no. 75.
[20] P. V. D. Corput and J. J. V. Wijk, "ICLIC: Interactive categorization of large image collections," in *Proc. IEEE Pacific Vis. Symp.*, 2016, pp. 152–159.
[21] M. A. Cox and T. F. Cox, "Multidimensional scaling," in *Handbook of Data Visualization*. Berlin, Germany: Springer, 2008, pp. 315–347.
[22] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.
[23] D. Deng et al., "Revisiting the design patterns of composite visualizations," 2022, *arXiv:2203.10476*.
[24] D. Deng et al., "VisImages: A fine-grained expert-annotated visualization dataset," *IEEE Trans. Vis. Comput. Graph.*, to be published, doi: 10.1109/TVCG.2022.3155440.
[25] V. Dibia and Ç. Demiralp, "Data2Vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks," *IEEE Comput. Graph. Appl.*, vol. 39, no. 5, pp. 33–46, Sep./Oct. 2019.
[26] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts, "Visual comparison for information visualization," *Inf. Visualization*, vol. 10, pp. 289–309, 2011.

orized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 28,2025 at 20:09:02 UTC from IEEE Xplore. Restrictions a

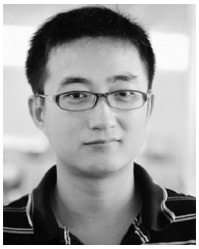
[27] S. R. Gomez, R. Jianu, C. Ziemkiewicz, H. Guo, and D. Laidlaw, "Different strokes for different folks: Visual presentation design between disciplines," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2411–2420, Dec. 2012.
[28] J. Han, J. Tao, and C. Wang, "FlowNet: A deep learning framework for clustering and selection of streamlines and stream surfaces," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 4, pp. 1732–1744, Apr. 2020.
[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[30] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley, "DNA visual and analytic data mining," in *Proc. IEEE Vis. Conf.*, 1997, pp. 437–441.
[31] E. Hoque and M. Agrawala, "Searching the visual style and structure of d3 visualizations," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 1, pp. 1236–1245, Jan. 2020.
[32] K. Hu, M. A. Bakker, S. Li, T. Kraska, and C. Hidalgo, "VizML: A machine learning approach to visualization recommendation," in *Proc. ACM CHI Conf. Hum. Factors Comput. Syst.*, 2019, pp. 1–12.
[33] K. Hu et al., "VizNet: Towards a large-scale visualization learning and benchmarking repository," in *Proc. ACM CHI Conf. Hum. Factors Comput. Syst.*, 2019, pp. 1–12.
[34] W. Javed and N. Elmqvist, "Exploring the design space of composite visualization," in *Proc. IEEE Pacific Vis. Symp.*, 2012, pp. 1–8.
[35] D. Jung et al., "ChartSense: Interactive data extraction from chart images," in *Proc. ACM CHI Conf. Hum. Factors Comput. Syst.*, 2017, pp. 6706–6717.
[36] H. Jänicke and M. Chen, "A salience-based quality metric for visualization," *Comput. Graph. Forum*, vol. 29, no. 3, pp. 1183–1192, 2010.
[37] M. C. Kim, Y. Zhu, and C. Chen, "How are they different? A quantitative domain comparison of information visualization and data visualization (2000–2014)," *Scientometrics*, vol. 107, no. 1, pp. 123–165, 2016.
[38] H. Li, Y. Wang, A. Wu, H. Wei, and H. Qu, "Structure-aware visualization retrieval," in *Proc. ACM CHI Conf. Hum. Factors Comput. Syst.*, 2022, pp. 1–14.
[39] H. Li, Y. Wang, S. Zhang, Y. Song, and H. Qu, "KG4Vis: A knowledge graph-based approach for visualization recommendation," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 1, pp. 195–205, Jan. 2022.
[40] Y. Liu, E. Jun, Q. Li, and J. Heer, "Latent space cartography: Visual analysis of vector space embeddings," *Comput. Graph. Forum*, vol. 38, no. 3, pp. 67–78, 2019.
[41] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[42] J. Luo, Z. Li, J. Wang, and C.-Y. Lin, "ChartOCR: Data extraction from charts images via a deep hybrid framework," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2021, pp. 1917–1925.
[43] Y. Luo, X. Qin, N. Tang, and G. Li, "DeepEye: Towards automatic data visualization," in *Proc. IEEE 34th Int. Conf. Data Eng.*, 2018, pp. 101–112.
[44] R. Ma et al., "LADV: Deep learning assisted authoring of dashboard visualizations from images and sketches," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 9, pp. 3717–3732, Sep. 2021.
[45] Y. Ma, A. K. H. Tung, W. Wang, X. Gao, Z. Pan, and W. Chen, "ScatterNet: A deep subjective similarity model for visual analysis of scatterplots," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 3, pp. 1562–1576, Mar. 2020.
[46] A. Mayorga and M. Gleicher, "Splatterplots: Overcoming overdraw in scatter plots," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 9, pp. 1526–1538, Sep. 2013.
[47] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*.
[48] W. Plant and G. Schaefer, "Visualisation and browsing of image databases," in *Multimedia Analysis, Processing and Communications*. Berlin, Germany: Springer, 2011, pp. 3–57.
[49] J. Poco and J. Heer, "Reverse-engineering visualizations: Recovering visual encodings from chart images," *Comput. Graph. Forum*, vol. 36, no. 3, pp. 353–363, 2017.
[50] C. Qian, S. Sun, W. Cui, J.-G. Lou, H. Zhang, and D. Zhang, "Retrieve-then-adapt: Example-based automatic generation for proportion-related infographics," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 443–452, Feb. 2021.
[51] O. Rooij, J. V. Wijk, and M. Worring, "MediaTable: Interactive categorization of multimedia collections," *IEEE Comput. Graph. Appl.*, vol. 30, no. 5, pp. 42–51, Sep./Oct. 2010.
[52] B. Saleh, M. Dontcheva, A. Hertzmann, and Z. Liu, "Learning style similarity for searching infographics," in *Proc. Graph. Interface Conf.*, 2015, pp. 59–64.
[53] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-Lite: A grammar of interactive graphics," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 341–350, Jan. 2017.
[54] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer, "ReVision: Automated classification, analysis and redesign of chart images," in *Proc. ACM 24th Annu. ACM Symp. User Interface Softw. Technol.*, 2011, pp. 393–402.
[55] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.
[56] H. J. Schulz, "TreeVis.net: A tree visualization reference," *IEEE Comput. Graph. Appl.*, vol. 31, no. 6, pp. 11–15, Nov./Dec. 2011.
[57] J. Sechler, L. Harrison, and E. M. Peck, "Sightline: Building on the web's visualization ecosystem," in *Proc. ACM CHI Conf. Hum. Factors Comput. Syst.*, 2017, pp. 2049–2055.
[58] Q. Shen et al., "StreetVizor: Visual exploration of human-scale urban forms based on street views," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 1004–1013, Jan. 2018.
[59] N. Siegel, Z. Horvitz, R. Levin, S. Divvala, and A. Farhadi, "FigureSeer: Parsing result-figures in research papers," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 664–680.
[60] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[61] S. Song, C. Li, Y. Sun, and C. Wang, "VividGraph: Learning to extract and redesign network graphs from visualization images," *IEEE Trans. Vis. Comput. Graph.*, to be published, doi: 10.1109/TVCG.2022.3153514.
[62] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.
[63] P. van der Corput and J. J. van Wijk, "Comparing personal image collections with PICTuReVis," *Comput. Graph. Forum*, vol. 36, no. 3, pp. 295–304, 2017.
[64] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.
[65] C. Wang et al., "Similarity-based visualization of large image collections," *Inf. Visualization*, vol. 14, no. 3, pp. 183–203, 2013.
[66] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. 2nd ed., Berlin, Germany: Springer, 2009.
[67] W. Willett, J. Heer, and M. Agrawala, "Scented widgets: Improving navigation cues with embedded visualizations," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1129–1136, Nov./Dec. 2007.
[68] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, no. 1–3, pp. 37–52, 1987.
[69] X. Xie, X. Cai, J. Zhou, N. Cao, and Y. Wu, "A semantic-based method for visualizing large image collections," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 7, pp. 2362–2377, Jul. 2019.
[70] J. Yang et al., "Semantic image browser: Bridging information visualization with automated intelligent image analysis," in *Proc. IEEE Symp. Vis. Analytics Sci. Technol.*, 2006, pp. 191–198.
[71] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst, "Faceted metadata for image search and browsing," in *Proc. ACM CHI Conf. Hum. Factors Comput. Syst.*, 2003, pp. 401–408.
[72] L. Yuan et al., "Deep colormap extraction from visualizations," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 12, pp. 4048–4060, Dec. 2022.
[73] W. Zeng, A. Dong, X. Chen, and Z.-L. Cheng, "VIStory: Interactive storyboard for exploring visual information in scientific publications," *J. Vis.*, vol. 24, no. 1, pp. 69–84, 2021.
[74] P. Zhang, C. Li, and C. Wang, "VisCode: Embedding information in visualization images using encoder-decoder network," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 326–336, Feb. 2021.
[75] B. Zhao and K. Han, "Novel visual category discovery with dual ranking statistics and mutual knowledge distillation," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2021, pp. 22982–22994.
[76] J. Zhao, M. Fan, and M. Feng, "ChartSeer: Interactive steering exploratory visual analysis with machine intelligence," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 3, pp. 1500–1513, Mar. 2022.

orized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 28,2025 at 20:09:02 UTC from IEEE Xplore. Restrictions a

Yilin Ye is currently working toward the PhD degree with the Hong Kong University of Science and Technology (Guangzhou). His current research interests include focuses on visualization and visual analytics, computational aesthetics, and HCI.



Rong Huang recieved the BArch degree in architecture from Hunan University, and the ME degree in architecture from Tongji University. She is currently working toward the PhD degree with the Hong Kong University of Science and Technology (Guangzhou). Her current research interests include computational aesthetics, urban renovation, and evidence-based design.



Wei Zeng (Member, IEEE) received the PhD degree in computer science from Nanyang Technological University, in 2015. He is an assistant professor with the Hong Kong University of Science and Technology (Guangzhou). He received ICIV'15 and VINCI'19 Best Paper Award, and ChinaVis'21 Best Paper Honorable Mention Award. He serves as Program Chair for VINCI'21, program committee for venues including IEEE VIS, EuroVis STARs, and ChinaVis. His recent research interests include visualization and visual analytics, computer graphics, AR/VR, and HCI.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.

rized licensed use limited to: The Hong Kong University of Science and Technology (Guangzhou). Downloaded on October 28,2025 at 20:09:02 UTC from IEEE Xplore. Restrictions a