InfoColorizer: Interactive Recommendation of Color Palettes for Infographics

Lin-Ping Yuan, Ziqi Zhou, Jian Zhao, Yiqiu Guo, Fan Du, Huamin Qu *Member, IEEE*

Abstract—When designing infographics, general users usually struggle with getting desired color palettes using existing infographic authoring tools, which sometimes sacrifice customizability, require design expertise, or neglect the influence of elements' spatial arrangement. We propose a data-driven method that provides flexibility by considering users' preferences, lowers the expertise barrier via automation, and tailors suggested palettes to the spatial layout of elements. We build a recommendation engine by utilizing deep learning techniques to characterize good color design practices from data, and further develop InfoColorizer, a tool that allows users to obtain color palettes for their infographics in an interactive and dynamic manner. To validate our method, we conducted a comprehensive four-part evaluation, including case studies, a controlled user study, a survey study, and an interview study. The results indicate that InfoColorizer can provide compelling palette recommendations with adequate flexibility, allowing users to effectively obtain high-quality color design for input infographics with low effort.

Index Terms—Color palettes design, infographics, visualization recommendation, machine learning.

1 Introduction

Infographics have been widely accepted as an effective means to convey abstract information to the general public. Besides the content and structure of infographic elements (e.g., shapes, pictograms, text, and indices), the colors of these elements and their combinations—*color palettes*—are essential, because they significantly influence the aesthetics, engagement, and memorability [21], [9]. However, either crafting an infographic or selecting an effective palette is not easy, especially for general users who lack expertise in design, since each task requires considering many factors simultaneously such as layout, appearance, and perceptual effectiveness. While many authoring tools [28], [34], [69], [64], [12], [15], [63] have been developed to facilitate infographics creation, these tools do not provide adequate color design support. Users are required to either manually craft color palettes or choose them among a predefined set.

Imagine a marketing manager, Linda, obtains a blue-background infographic online, and wants to use it in her slides with the company brand theme, which is red. She loads the infographic into Adobe Illustrator, but soon gets stuck in attempting to create a color palette from scratch. While there are many principles for color design, Linda is not familiar with them, and thus has no idea how to leverage them to get a harmonious palette. Thus, she turns to the predefined palettes in the tool, but finds limited available choices to satisfy her needs. She wants the background red while having some elements' colors to reflect affective or semantic information. Even a palette meeting all the requirements is finally found, there is still a big question on which color in the palette should be applied to which element of the infographic. The spatial layout of these elements matters [44], [33], for example, a piece of text of less contrast color with its background element is hard to read.

The above example reveals three key common challenges of designing color palettes using the existing tools: 1) creating a palette from scratch requires users having relevant expertise, 2) using predefined palettes by the tool limits users' freedom, and 3) applying a palette to an infographic is complicated due to the spatial layout of elements.

To address these challenges, we propose *InfoColorizer*, an interactive tool that allows general audience to effectively design color palettes during infographic creation, using a data-driven approach (Fig. 1). We employ deep learning to extract color design practices from a large dataset of infographics created by designers, and then use the learned model to recommend appropriate color palettes. This *lowers the expertise barrier* of users to craft good color palettes.

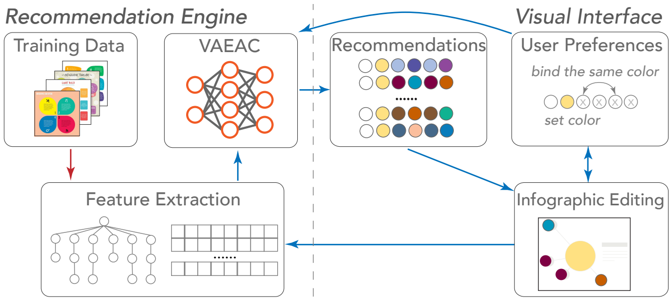

Fig. 1. InfoColorizer consists of a recommendation engine and a visual interface. The recommendation engine is constructed by first extracting structure and color information from inforgraphics and then training a deep learning model, VAEAC, to characterize good color design practices. The visual interface further allows users to obtain recommended palettes, specify various preferences, edit infographics, and retrieve new recommendations iteratively.

- *Lin-Ping Yuan and Huamin Qu are with the Hong Kong University of Science and Technology. E-mails: {lyuanaa, huamin}@cse.ust.hk.*
- *Ziqi Zhou and Jian Zhao are with the University of Waterloo. E-mails: {z229zhou, jianzhao}@uwaterloo.ca.*
- *Yiqiu Guo is with the Xi'an Jiaotong University. Email: maxleaf@stu.xjtu.edu.cn.*
- *Fan Du is with Adobe Research. Email: fdu@adobe.com.*
- *Jian Zhao is the corresponding author.*

Manuscript received January 18, 2021; revised May 11, 2021.

Particularly, we frame the learning process as a conditional generative problem, and leverage VAEAC (Variational AutoEncoder with Arbitrary Conditioning) [25] to recommend color palettes dynamically based on conditions (e.g., color preferences) set by users. This *offers flexibility* to users by enabling partial specification of palettes with exact or vague color constraints. Moreover, we characterize infographics with features including information about the elements' spatial layouts in the dataset, allowing for integrating such knowledge into our learned model. This suggests color palettes *tailored for particular element arrangements* in infographics. InfoColorizer also supports some basic editing functions, allowing users to try out different infographic layouts, obtain corresponding palette recommendation, and iteratively refine their design.

We validated InfoColorizer through a comprehensive evaluation comprising four parts. First, we demonstrate the usefulness of InfoColorizer with case studies using real-world infographics and example scenarios. These cases reveal that the system can facilitate color palettes design with cogent recommendations in different tasks such as filling empty wireframes and combining infographics with different color schemes. Then, we conducted a controlled user study with 24 design novices. The qualitative and quantitative results show that InfoColorizer offers higher efficiency and better creativity support than a baseline with manual color design and online resources. Third, we carried out an online survey study with 102 users to compare artist-designed, predefined, randomly-generated, baseline-crafted and InfoColorizer-recommended color palettes on aesthetics and readability. The results indicate that although InfoColorizer's recommendations were not perceived to be as good as artist-designed palettes, they received higher scores than the other three methods on both factors. Finally, we interviewed four graphic design experts in depth; they appreciated InfoColorizer's novel features and were able to generate compelling infographics meeting their needs effectively within a few operations. In summary, our main contributions include:

- A novel data-driven approach that recommends palettes for infographics by leveraging deep learning techniques with the consideration of elements' spatial arrangements, while offering flexibility for user preferences of colors;
- An interactive tool, InfoColorizer, that incorporates the data-driven recommendation and makes it easily accessible and manageable to users, along with the support of iterative design and basic infographic editing; and
- Insights and results from a series of evaluations covering case studies, a controlled user study, an online survey, and an interview study.

The source code of the system (including the models, user interface, examples, and appendices) is available at https://github.com/yuanlinping/InfoColorizer.

2 Related Work

2.1 Infographic Models and Authoring Tools

Compared to plain charts or text, infographics incorporate additional visual embellishments (e.g., icons, images, and pictograms) to convey abstract information appealingly. Previous empirical studies have shown that they can increase comprehension and engagement without reducing viewers' accuracy [8], [20]. Further, researchers have demonstrated that colors are essential to make visualizations memorable and influence a first impression [9], [21], [51].

Besides empirical studies, data-driven methods or machine learning models have been applied for estimating the visual importance [11], exploring perceived personalities [71], and computing similarity of infographics to facilitate search [48]. Recently, Lu et al. [36] explored high-level narrative flows extracted from infographic collections. However, none of them have focused on the color aspect of infographics design.

To facilitate the creation of infographics, researchers have developed many authoring tools, which fall into three main categories: programming toolkits, interactive design environments, and semi- or fully-automated tools. Programming toolkits (e.g., D3 [10]) allow users to create visualizations with the greatest extent of flexibility. However, they usually have a steep learning curve and are challenging for general users. Thus, many interactive design environments have been developed for users without programming skills, such as Adobe Illustrator. Also, tools that support data binding have been proposed to ease the creation of data-driven charts and infographics, including the binding of standard marks [49], icons [64], and hand-drawn shapes [28], [69], [34]. However, they still require users to manually craft infographic elements. To further lower the barrier, semi- or fully-automatic tools were developed to support creating infographics directly from different inputs, such as natural language statements [15], tables [63], graphs [57], and timeline images [12].

While providing rich capabilities, almost all the above systems leave users with two options to obtain a palette—which is critical for the aesthetics of their infographics—including: 1) manually creating one from scratch and 2) choosing from a predefined collection. The former requires expertise in color design and the latter limits a user's flexibility. Our work aims to address these issues via an interactive recommendation. Motivated by the discussion of Lu et al. [36] on the spatial structure of infographics, we go a step further to integrate the information of elements layout into suggesting proper palettes.

2.2 Color Palette Design Principles and Tools

Prior studies mainly focus on improving perceptual discriminability and aesthetics of designed palettes in data visualization. Levkowitz et al. [31] suggested that discriminable palettes should effectively mirror encoded data and accurately convey data differences. Visualization designers are recommended to consider many factors, such as underlying data types [58], visual consistency [45], tasks [59], and color properties (from basic visual separability [62], contrast [40], and difference metrics [56], to more advanced color appearance [30], name differences [23], affect [7], and semantics [32], [50]). Moreover, studies indicated that using harmonious colors [39] or adjusting hue and saturation [41] can increase the aesthetics of visualizations.

However, applying these principles in practice can be difficult for non-experts. Many techniques have been proposed to ease this process, such as ColorBrewer [22] that provides predefined palettes for encoding sequential, diverging, and qualitative data. In the case of graphic design (i.e., not encoding any data), some commercial interactive tools (e.g., Adobe Color [6]) utilize harmonic templates to help users craft customized palettes with high quality.

Further, researchers have developed algorithms to automatically generate palettes for different applications. For categorical data, Colorgorical [19] creates palettes based on user-defined balance of color discriminability vs. aesthetic preference. Palettailor [35] further provides a data-aware approach that generates and assigns colors for multi-class scatterplots, bar and line charts to maximize their visual discrimination. For numerical data, Smart et al. [54] modeled expert-designed color ramps to suggest palettes. Besides data charts or visualizations, algorithms have been designed for (re)coloring other types of media, such as natural images and patterns [14]. O'Donovan et al. [43] proposed a color compatibility model that can score the quality of any five-color palettes. By considering compatibility and spatial arrangements of colors, Lin et al. [33] suggested a probabilistic model for coloring 2D patterns.

However, the above techniques focus on charts/ visualizations, natural images, or patterns; none of them are developed for infographics with unique characteristics. First, infographics consist of both data elements and visual embellishments, where colors play multiple roles such as visual group indicators and artistic decorators [36]. Second, the spatial arrangement of its elements is different from that in other media, which may exhibit complicated semantics and convey a narrative. Perhaps the general methods on suggesting palettes for website design [6], [43], [22], [19] can be used for infographics. However, they are limited in generating palettes with a fixed number of colors (e.g., five), and do not support assigning colors to infographic elements.

2.3 Visualization Recommendation

Researchers have explored various techniques for recommending appropriate visualizations, including some commercial tools such as Tableau. One category falls in rule-based methods. APT [38] introduces a compositional algebra to enumerate the space of charts and ranks them, which was later extended in SAGE [47]. CompassQL [65], the basis of Voyager [66] and Voyager 2 [67], offers flexible query specifications for searching the visualization space and providing recommendations. Further, Draco [42] leverages answer set programming to describe constraints over visualization design. Another category is data-driven, based on machine learning techniques. VizML [24] learns design choices from a corpus of data and visualization pairs. Data2Vis [17] is an end-to-end neural network that generates visualizations directly from data. DeepEye [37] and LQ2 [68], on the other hand, combine rule-based methods and machine learning to rank and classify visualizations.

While the above systems allow users to effectively create visualizations from input data, none of them adequately support recommending color designs of generated charts. Moreover, as mentioned above, infographics have unique characteristics that are different from ordinary charts or visualizations, which is the focus of our work.

3 InfoColorizer Design and Overview

In this section, we outline the design goals for developing InfoColorizer, followed by an overview of our method.

3.1 Design Goals

Motivated by the aforementioned scenario and limitations of the existing tools, we derive the following design goals.

G1: Lower expertise barrier for crafting professional palettes. Creating high-quality palettes is challenging for general users as many factors need to be considered simultaneously, such as aesthetics, harmony, and perceptual discriminability. *The system should characterize good practices embodied in handcrafted designs, and assist users with automatic palette recommendation that reflects these good practices.*

G2: Offer flexibility to embed different kinds of user preferences. Users may have constraints when creating palettes, such as applying a specific or semantically meaningful color to a particular element, and setting multiple relevant elements with the same color for consistency. *The system should provide a flexible mechanism to allow for specifying various types of color preferences on elements of interest, as well as alleviate users from considering colors for other elements.*

G3: Incorporate consideration of spatial arrangements of elements. A good palette may work poorly with improper color assignments because elements' spatial layout influences color appearance. However, determining an assignment from numerous possibilities (e.g., around 5^{10} assignments for a five-color palette and a ten-element infographic) requires much trial-and-error tweaking. *The system should adapt palettes to particular spatial arrangements of input infographics in recommendations, thus freeing users from the tediousness of tuning color assignments.*

G4: Support simple user interactions and iterative design of color palettes. General users rely on interfaces to access system functions. *The system should provide intuitive user interactions such as obtaining palette recommendations, specifying color preferences, and previewing & editing infographics.* Further, the palette design process is often iterative by trying different ideas. *The system should facilitate refining results in a human-in-the-loop manner, such as bookmarking recommended palettes, and tuning constraints according to their needs.*

3.2 Method Overview

As shown in Fig. 1, we employ a data-driven approach to automatically acquire good practices exhibited in infographic collections and then utilize the "learned knowledge" to recommend palettes, with a visual interface that allows interactions with the recommendation engine.

More specifically, considering an infographic $\mathbf{I} = \{E_1, E_2, \cdots, E_n\}$, where E_i is an element, we characterize $\mathbf{I}$ with a set of non-color features $\mathbf{F} = \{F_1, F_2, \cdots, F_m\}$ and color features $\mathbf{C} = \{C_1, C_2, \cdots, C_n\}$ for the n elements (see Sec. 4). The non-color features $\mathbf{F}$ include information at different granularity (e.g., infographic and element levels), and the spatial arrangement of elements, which are combined and represented in a tree structure (G3). For expert-designed infographics, the color features $\mathbf{C}$, and their relations with the features $\mathbf{F}$ reflect good practices that we wish to capture.

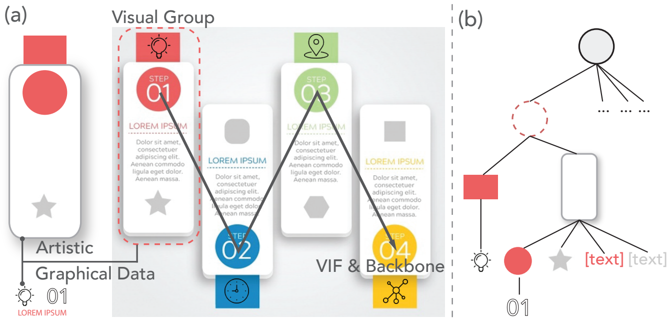

Fig. 2. (a) Artistic elements, graphical data elements, visual groups, and visual information flow (VIF) & backbone in an infographic. (b) The conceptual tree model. Some branches are omitted for simplicity.

We therefore frame our recommendation process as a conditional generative problem (see Sec. 5). We employ Variational AutoEncoder with Arbitrary Conditioning (VAEAC) [25] as our generative model, because of its flexibility in adapting any features as conditions. That is, given a collection of expert-designed infographics, with features $(\mathbf{F}_k, \mathbf{C}_k)$, the model can learn a probability distribution over the full feature set—non-color features $\mathbf{F}$ and colors $\mathbf{C}$—to capture the good practices (G1). Later, the learned model can be used to generate any "missing" features of an infographic $\mathbf{I}$ with knowing the rest (i.e., the arbitrary conditions). For example, users can specify colors C_i and C_j for certain elements E_i and E_j, and the conditional generative problem becomes sampling from $p(\mathbf{C} \setminus C_{i,j} | \mathbf{F}, C_{i,j})$, allowing for the flexibility of incorporating different kinds of user preferences (G2). To make the above recommendation easily accessible and configurable, we design a visual interface for InfoColorizer, which also enables iterative generation of colors and simple infographic editing functions (G4).

4 Dataset and Infographic Model

To achieve the design goals, the first step is to identify high-quality infographic datasets from which a data-driven method can extract good palette design practices (G1). Further, we need to conceptually model infographics in a form that is effective for algorithms to understand and process.

4.1 Dataset

Previous studies collected several infographic datasets, such as MassVis [3], Visually29K [5], InfoVIF [36] and Timelines [12]. We chose InfoVIF as our initial test bed because: 1) compared to MassVis and Visually29K, InfoVIF tends to be more useful for general audiences and has more uniform styles of visual elements and layouts, allowing machine learning to better capture common design patterns; 2) compared to Timelines, InfoVIF covers a wider range, including not only timelines but also other types; 3) infographics in InfoVIF are contributed by world-wide designers with high-quality and diverse design themes. Thus, InfoVIF is a suitable resource from which good color design practices can be extracted (G1).

4.2 Conceptual Model of Infographics

As shown in Fig. 2-a, Lu et al. [36] characterized an infographic as a combination of *artistic decorations* (e.g., shapes, images, and cliparts) and *graphical data elements* (e.g., icons, text, and indices). Graphical data elements are then organized into *visual groups* to convey pieces of information. Visual groups are further presented in sequential order, called *visual information flow (VIF)*, to indicate a narrative of the infographic. The path connecting the semantic centroids of the visual groups is called VIF *backbone*.

However, this infographic model is not sufficient for our case, because it *only* considers graphical data elements when composing visual groups. This is oversimplified, because the artistic decorations and their colorfulness largely affect the aesthetics of infographics, and thus are important in determining proper color palettes for a visually compelling infographic. Further, while VIF captures a high-level spatial structure to make logical sense, more detailed spatial relationships between individual elements arguably influence the color palette design. For example, two elements next to each other (i.e., *adjacency*) may share the same color to imply the "Similarity" Gestalt Law [16]; and one element fully on top of another (i.e., *inclusion*) is benefited from choosing a high contrast color with respect to the one below.

To address these issues, we first extend Lu et al.'s model [36] by including both artistic and graphical data elements in visual groups, as well as VIFs, which characterizes an infographic from a logical perspective (Fig. 2-a). Inspired by trees being used to analyze topological structures of images [55], we further employ a tree structure, which incorporates Lu et al.'s model and characterizes an infographic from a spatial perspective (G3). The structure can describe the adjacency and inclusion spatial relationships at multiple levels (Fig. 2-b). The root node of a tree represents the whole infographic (i.e., the background canvas), and the second layer of the tree represents all the visual groups, whose descendant nodes are artistic or graphical data elements within the corresponding visual group. Under a visual group node, a parent-child link indicates element inclusion, and a sibling relation indicates adjacency in layout. Our model conceptualizes an infographic from both logical and spatial aspects, allowing data-driven models to extract associations between these aspects and color designs.

5 InfoColorizer System

In this section, we describe InfoColorizer in detail. As shown in Fig. 1, the system consists of a recommendation engine and a visual interface. We first describe the feature extraction process (Sec. 5.1) and how we characterize good color design practices using the VAEAC model (Sec. 5.2). We then demonstrate InfoColorizer's ability to support flexible color preference specifications (Sec. 5.3), and the visual interface that enables an effective palette design workflow (Sec. 5.4).

5.1 Characterize Infographics with Various Features

For an infographic $\mathbf{I}$, we compile a set of color $\mathbf{C}$ and non-color $\mathbf{F}$ features to describe its visual appearance and spatial arrangement of elements (G3).

We extract the color values of all elements and the infographic background as the color features $\mathbf{C}$. CIELab color space is used because of its perceptual uniformity. We distill a list of non-color features $\mathbf{F}$ at the multiple levels. At **infographic level**, we obtain *VIF Type*, *Visual Group Number*, and *Visual Group Distance*. Features in **visual group level** contain *Visual Group Element Number* and *Relative Visual Group Size*. At **element level**, we extract *Element Type*, *Relative Element Size* and *Relative Element Pixel Area* for each artistic and graphical data element. In addition, we use the nested set model [18] to represent the **tree structure** (Sec. 4.2) of an infographic by storing *Left Index Number and Right Index Number* of each node. A complete explanation of the above features and technical details about the extraction algorithms can be found in our online Appendix Sec. A.

The non-color features embed many design factors considered by experts. Specifically, they help organize information about narrative flow (*VIF*), visual groups (*Visual Group Number and Distance*), size(*Relative Visual Group Size, Relative Element Size/Pixel Area*), shape (*Element Type*). Besides, the tree structure reveals the spatial arrangement of elements (G3) and other design factors, such as layers by tree traversal background/foreground contrast by inferring parent-child relationships, etc. Characterizing these factors makes capturing high-level color design practice from infographics possible with a data-driven approach.

5.1.1 Technical Details for Features Extraction

While Lu et al. [36] provided methods for extracting VIF and graphical data elements, our key technical challenges include identifying artistic elements and constructing the tree structure as described in Sec. 4.2.

Infographic Level Features Extraction. We employ the data element extraction and VIF construction algorithms in [36]. Their data element extraction utilizes a state-of-art object detection model to identify the bounding boxes of graphical data elements (e.g., icons, text, indices) in an infographic. Based on the detected elements, the VIF construction algorithm leverages Gestalt principles to identify the visual groups and VIF backbone. Therefore, we can easily compute the `VIF Type`, `Visual Group Number`, and `Visual Group Distance`.

Artistic Elements Identification. The algorithms in [36] can only detect graphical data elements, whereas identifying artistic elements is essential for us to compute the features at the visual group and element levels and to construct a precise tree model of an infographic. An intuitive idea is to find areas with the same (or similar) colors using color segmentation [60], because an artistic element is usually exhibited as a shape with a consistent color or smooth color gradient. We achieve this via three main steps (Fig. 3):

1) *Remove graphical data elements.* The graphical data elements can interfere with our color segmentation, as they are also colored and often overlay on top of artistic elements. To remove a data element (Fig. 3-b), we simply set all the pixels within the bounding box with its background color (usually the color of the artistic element below or the infographic background).
2) *Perform color segmentation.* Then, we group continuous pixels with similar colors (in CIELab color space) using a region growing algorithm [60] (Fig. 3-c). A threshold in CIEDE2000 color difference needs to be set, and we experimentally found that setting the threshold to 4 yields good segmentation results.
3) *Merge segments with similar color hues.* An artistic element may contain a color gradient, resulting in multiple segments from the previous step. As the segments usually have similar hues, we apply kernel density estimation (KDE) clustering [27] on hues and group segments in the same cluster into one (Fig. 3-d). We used Gaussian kernel and set the bandwidth to 3.

We therefore obtain a continuous region of pixels that represents an artistic element, and thus can easily compute its bounding box.

Tree Construction. With all the graphical data and artistic elements identified, we now can construct a tree structure described in Sec. 4.2 based on their bounding boxes. We start by considering each element as a node, and construct the tree from top to bottom. An edge is added between two elements if one's bounding box directly contains the other's without others spatially in-between. As shown in Fig. 4-a, we then obtain a tree whose root node is the background canvas of an infographic and other nodes are either graphical data elements (in blue strokes) or artistic elements (in black strokes). Next, we group branches containing graphical data and artistic elements within a visual group (based on the VIF construction algorithm), and insert visual group nodes (in green strokes) below the root (Fig. 4-b). With this tree representing the logical structure as well as the spatial arrangement of elements in an infographic, we conduct a pre-order traversal on the tree and compute the `Left Index Number` and the `Right Index Number`.

Visual Group Level and Element Level Features Extraction. After identifying all the elements and visual groups with the above procedures, most of the visual group and element level features (such as `Visual Group Element Number`, `Relative Visual Group Size`, `Relative Element Size`, and `Relative Element Pixel Area`) are easy to compute. The `Element Type` of graphical data elements can be obtained by the data element extraction in [36]. To recognize the `Element Type` of an artistic element, we first find its contour (consisting of pixel locations) using Suzukil et al.'s algorithm [55]; and then we compute an approximated contour (consisting of a few vertices) using the Ramer-Douglas-Peucker algorithm [46]. We classify the artistic elements based on their shape using the vertices in the approximated contour. For example, if there are three vertices, we recognize it as a *triangle*.

5.2 Lower Expertise Barrier with Recommendation

The next step is to train a machine learning model that extracts good design practices in data and utilizes that for providing recommendations (G1). As mentioned in Sec. 3.2, we frame the recommendation process as a conditional generative problem, and employ Variational AutoEncoder with Arbitrary Conditioning (VAEAC) [25], a special kind of Variational AutoEncoders (VAE) [29].

In general, a VAE aims to learn a *bidirectional* mapping between data items x in an application domain and variables z in a continuous latent space. The model consists of two parts—an encoder $p(z|x)$ that converts x to z and a decoder $q(x|z)$ that does the opposite; and the training process is to learn the two probability distributions. In our case, x is the infographic features, $[\mathbf{F}, \mathbf{C}]$, and z reflects the abstract knowledge in data.

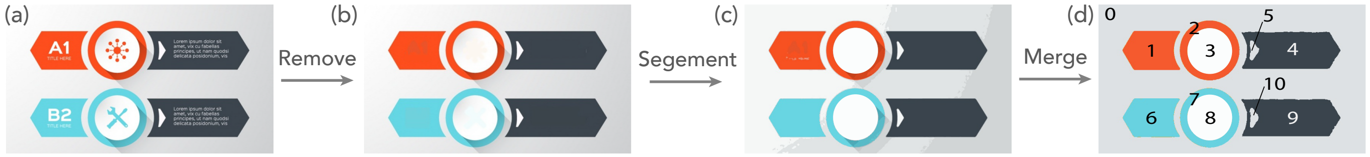

Fig. 3. Three steps to identify artistic elements in an infographic: removing graphical data elements, performing color segmentation, and merging segments with similar hues.

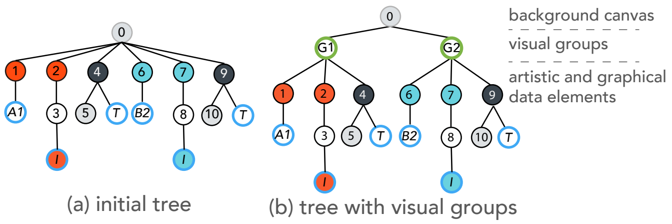

Fig. 4. Constructing the conceptual tree model of the infographic in Fig. 3. Blue, black, and green stroked circles indicate graphical data elements, artistic elements, and visual groups, respectively (*T* denotes *Text* and *I* denotes *Icon*).

We want to obtain color palettes according to a specific design of infographic reflected in $\mathbf{F}$, which is a *conditional* generation. Moreover, users may have specific preferences on coloring certain infographic elements (G2), meaning that some part of $\mathbf{C}$ is in the conditions. VAEAC, as an extension of VAE, can handle the above requirements, by incorporating a binary mask vector b that controls which part of x is the conditions (*observed* features) or not (*unobserved* features). Its encoder and decoder are $p(z|x_{1-b}, b)$ and $q(x_b|z, x_{1-b}, b)$, respectively, where x_{1-b} is the observed part and x_b is the unobserved part. The model is trained with a full set of features x, and can fill in any "missing" unobserved part of x (i.e., called *feature imputation*) during the generation stage with a dynamic input of b. When applying to our case, we treat $\mathbf{F}$ always observed and set part of $\mathbf{C}$ observed, controlled by a user input b. If none of $\mathbf{C}$ is observed (i.e., $b = \vec{1}$), the model can generate a full color palette for an infographic; and if some colors of $\mathbf{C}$ is observed (i.e., specified by a user with certain colors), the model can generate the rest of a color palette with these colors satisfying users' constraints. Details regarding the user preferences support will be introduced in Sec. 5.3.

Fig. 5 illustrates the idea of conditional generation behind VAEAC with an example of how spatial arrangements influence the recommended colors. We trained a VAEAC model on the obtained features $[\mathbf{F}, \mathbf{C}]$ described in Sec. 5.1 and another VAEAC model on the same dataset but removing spatial features representing tree structures (e.g., *Left and Right Index Number*). Then, we recommended 200 palettes with each trained model for each of the two infographics shown in Fig. 5. The two infographics have four visual groups, each consisting of a circle and an annular sector, whose spatial relationships are either inclusion (Fig. 5-a) or adjacency (Fig. 5-b). Next, for each combination of the two infographics and models, we calculated the distances between recommended colors of the circle and annular sector in the upper left visual groups (highlighted with green strokes) and obtained their distributions (Fig. 5-c) with kernel density estimation. We observe that the distributions generated by VAEAC trained with spatial features (blue lines) are distinct, while the distributions generated by VAEAC without spatial features (red lines) are similar. This indicates that the extra spatial information enables VAEAC to distinguish spatial arrangements and generate palettes tailored to different cases. Specifically, the model trained with spatial features tends to recommend similar and dissimilar colors for adjacent and overlapping elements, respectively. This aligns with the practice in the training dataset, in which designers use close colors to indicate adjacent elements within a visual group, and different colors to separate overlapping elements.

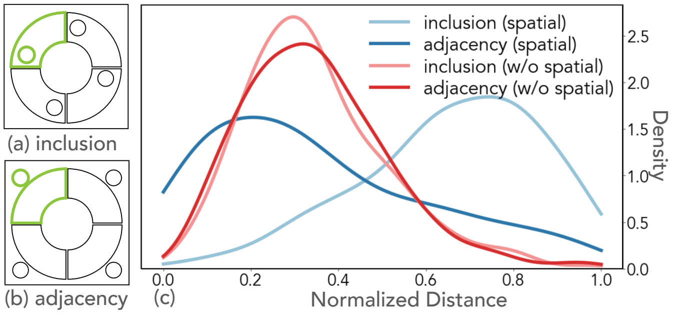

Fig. 5. Distributions (c) over distances between colors predicted by VAEAC models trained with or without spatial features for elements with inclusion (a) and adjacency (b) relationship.

To further investigate the performance of VAEAC, we also considered two alternative models including: 1) Generative Adversarial Imputation Nets [70], which is the state-of-the-art GAN-based model for feature imputation, and 2) Multivariate Imputation by Chained Equations [61], which is a classic non-deep-learning method. Our experiments indicated that VAEAC performed the best. Details about the model training, comparison, and evaluation can be found in the online Appendix Sec. B.

5.3 Offer Flexibility with Versatile User Preferences

With a trained VAEAC model, InfoColorizer can not only recommend colors for each infographic element but also support flexible control over the colors in recommendation (G2). InfoColorizer supports two kinds of user constraints: 1) specifying colors for certain elements in either an exact or a vague form, and 2) binding several elements together to have the same color in recommended palettes.

As mentioned in Sec. 5.2, VAEAC can generate unobserved colors conditioning on non-color features and observed colors. Thus, we can generate palettes meeting users' preferences by manipulating the input feature vector. Specifically, if a user assigns an exact color (e.g., in CIELab space $[l, a, b]$) to an element E_i, the corresponding color features of E_i are set to $[l, a, b]$, indicating these features are observed. Moreover, users can assign colors to an element E_i semantically using a word (referring to a range of colors) such as a *color name* (e.g., red, skyblue), an *object* (e.g., apple, dollar), and a kind of *affect* (e.g., exciting, calm). To handle such vague specifications, we first collected over 200 *(word, colors)* pairs from the previous works on color names [23], color affect [7], and color semantic [32]. We then utilize this information to manipulate the input feature vectors. When a word is assigned to an element E_i, we randomly select k colors from the corresponding color set and then generate k input vectors (we set k to 3 in InfoColorizer), where each has a different color (from $[l, a, b]_1$ to $[l, a, b]_k$) for the observed features of E_i. We can thus obtain k sets of recommendations and randomly pick some for presentation.

Users can also bind relevant elements (e.g., those within a visual group, or all icons, text, etc.) to constrain them with the same color in recommendations. We adopt a post-processing method on the recommended color palettes. For example, suppose that E_i, E_j, E_k are bound, for each recommended color palette, we randomly select one of them based on a probability decided by their areas, and then set all three elements with the color of the selected one.

5.4 Support User Workflow with Visual Interface

We develop a visual interface that enables users to iteratively obtain desired palettes by supporting basic editing and previewing functions, color preferences and other constraint specifications, and interactive recommendation (G4).

The interface (Fig. 6) consists of three interactively-coordinated panels. The *Content Library* (Fig. 6-A) stores raw materials (e.g., shapes, images, icons) and infographic templates. Users can create an infographic either from scratch or based on a template, and edit it on the *Main Canvas* (Fig. 6-B). The toolbar on the top supports some simple editing functions such as arrange, group/ungroup, duplicate, and delete. The *Control Panel* (Fig. 6-C) is a core component, where users can obtain desired palettes by iteratively specifying preferences, obtaining recommendations, and refining the design. Overall, we designed the interface with common panels and components to improve the usability and learnability. However, we proposed a novel widget for setting color preferences (Fig. 6-C1) with visualizations of layered elements and interactive linking.

Here, we use a simple scenario to demonstrate an interactive workflow of obtaining desired palettes with InfoColorizer. Suppose the marketing manager, Linda, wants to improve a chosen infographic from the Content Library (the first one in Fig. 6-A). She then clicks "Analyze Infographic" which analyzes the spatial relationships of its elements and displays a tree structure on the Color Preferences section (Fig. 6-C1). The tree, informing our conceptual infographic model (see Fig. 2), is visualized as rectangles in horizontal layers. The bottom layer is the background canvas, and the second bottom layer contains the elements directly placed on the background, and so forth. The color of a rectangle indicates users' preferences of the corresponding element; an empty rectangle with a diagonal line means no constraints.

Initially, she wants the background to be lighter and all the text to be pure white consistently. With the Color Preferences section (Fig. 6-C1), she assigns a vague color specification with the word "light" to background canvas; the word is shown on top of the rectangle. She then binds the four text objects and sets "#FFFFFF" (i.e., white) as their colors. The bound elements, which will always have the same colors in recommendations, are indicated by the small red dots below. The resulting color preference setting is shown in Fig. 6-①-CP1.

Linda clicks "Get Recommendations", and a list of recommended palettes meeting her needs are then returned by InfoColorizer (Fig. 6-④). The number of returned recommendations can be adjusted in InfoColorizer (the default is five). She picks her favorite one, Fig. 6-④-P1, for preview and refinement. The chosen palette is then duplicated in the Color Preferences section (Fig. 6-①-CP2), and the infographic is automatically colored by the palette (Fig. 6-⑤).

However, Linda is not satisfied with the colors of the first and last bars (i.e., "01" and "05"). She thus clears the colors of the two bars (Fig. 6-①-CP3), and requests new recommendations with this preference setting. The results are shown in Fig. 6-③. Similar to the previous iteration, she picks her favourite palette in Fig. 6-③-P2, which updates the Color Preference section (Fig. 6-②) and the infographic (Fig. 6-⑥). Linda is quite happy with this color design and exports the infographic for her presentation slides.

6 Evaluation

To assess the effectiveness and usefulness of InfoColorizer, we conducted a four-part evaluation. We first used several case studies to demonstrate that InfoColorizer can generate compelling palettes under different scenarios. In addition, we quantitatively and qualitatively evaluated InfoColorizer from the perspectives of novice creators, infographic readers, and graphical design experts separately by conducting a controlled user study, a survey study, and an interview study. These studies comprehensively reflect the strengths and weaknesses of InfoColorizer from different aspects.

6.1 Case Studies

Fig. 7 presents a set of infographics colored by recommended palettes. More examples can be found in online Appendix Sec. C. We demonstrate how InfoColorizer can facilitate palette creation under different user preferences and constraints. We consider three use cases: a) colorizing a *wireframe* infographic, b) improving the color *readability* of an infographic, and c) *stitching* two infographics with different color schemes. For each case, we select one infographic from a website [4]. We demonstrate InfoColorizer's recommendations under four conditions: 1) *no preferences*, 2) *exact color* specification, 3) *vague color* specifications, and 4) *element binding*. In Fig. 7, we use "Pin" icons to indicate elements that are specified with exact colors, annotate the words on elements that are specified vaguely, and add links to elements that are bound together. We can see that InfoColorizer can generate compelling palettes for the source infographics under different conditions; all of the results are obtained through one to two requests of recommendations.

Even without any color preferences, InfoColorizer is able to suggest cogent color palettes. For example, in Fig. 7-A1, the colors of the four shapes are in a smooth and consistent theme. In Fig. 7-C1, a user wants to obtain a palette for the head adapted to the bulb, and thus sets no preferences for the head while fixing colors for the bulb. The system recommends a color palette for the head similar to that of

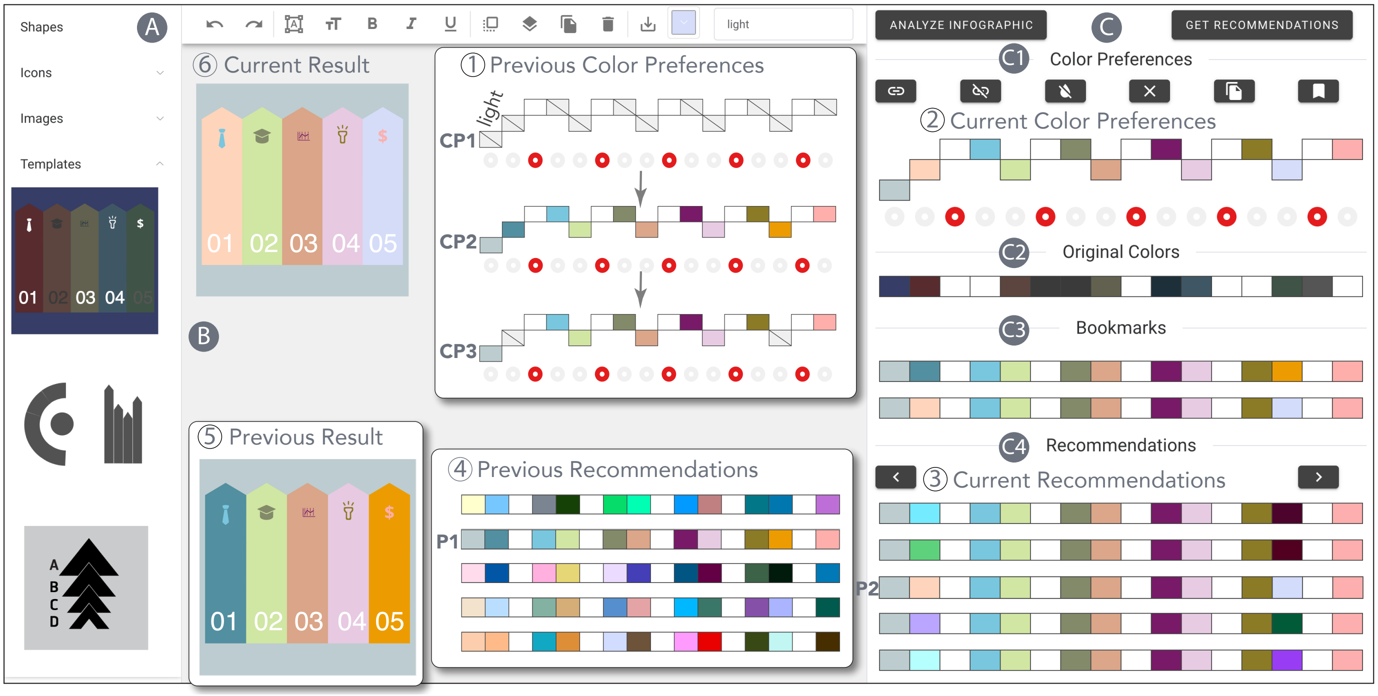

Fig. 6. InfoColorizer consists of three linked panels: (A) Content Library stores materials for creating infographics; (B) Main Canvas supports simple editing functions to compose infographics; (C) Content Panel offers setting color preferences and viewing recommendations, composed of four sections: (C1) Color Preferences, (C2) Original Colors, (C3) Bookmarks, and (C4) Recommendations. This allows users to obtain desired color palettes by iteratively ①② specifying preferences, ③④ getting recommendations, and ⑤⑥ previewing the results for further refinement.

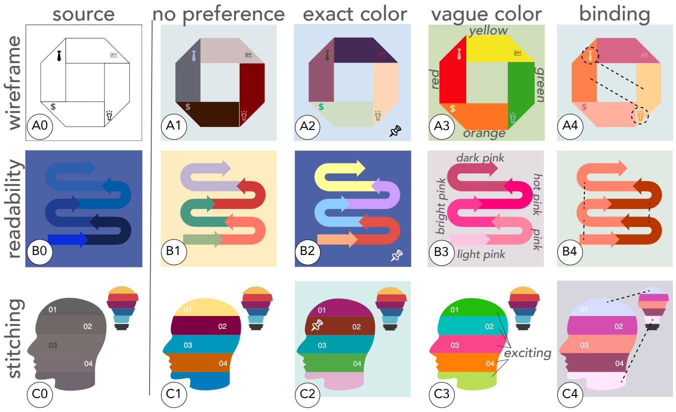

Fig. 7. Example cases of three different infographic creation scenarios with four different user preference conditions.

the bulb, indicating that the model utilizes *observed* color features (i.e., the bulb) to influence *unobserved)* color features (i.e., the head). Moreover, by assigning the backgrounds in Fig. 7-A2, B2 with desired colors, the original foreground colors are filled or enhanced with colors in contrast to the backgrounds. Further, after specified with some words of general color hues and semantics, InfoColorizer returns aesthetically pleasing infographics. For example, in Fig. 7-B3, a user demands a pink-themed infographic but has no idea about what specific pink colors are the best, and thus specifies general color categories with words. In Fig. 7-C3, "exciting" is attached to all the color strips on the head to match the underlying semantics of getting an idea (i.e., the bulb). Finally, by binding some elements together, a user can obtain more consistent color palettes, such as linking the corresponding arrows in Fig. 7-B4 to indicate different types of stages, and associating the relevant parts between the head and the bulb in Fig. 7-C4.

6.2 Controlled User Study

We further conducted a controlled study to evaluate the effectiveness of InfoColorizer with real users. This study aimed to investigate two aspects of the workflow: **(S1)** whether InfoColorizer can facilitate users for obtaining satisfactory palettes for their infographics, and **(S2)** whether InfoColorizer can support users' creativity in designing palettes. In particular, we investigate these questions under the situations that users have specific color preferences and consider the spatial layout of graphic elements.

6.2.1 Study Setup

Baseline. We considered a baseline to simulate a common palette design process in practice, in which people derive palettes from different sources (e.g., color pickers, predefined palettes, and online websites), and then color their infographics with a design tool [26]. We thus developed a baseline version of InfoColorizer by disabling the palette recommendation function and allowing users to seek colors via online resources and tools. In particular, we suggested three widely-used websites: Adobe Color [6], ColorBrewer [1], and Coolors [2]. They allow users to explore numerous expert-designed or automatically generated palettes, craft palettes from scratch using harmony rules, and search palettes with words like *lucky*, *moonlight*, covering a range of functions offered in InfoColorizer. However, it is not mandatory for users to use all of them. Instead, users can use any number of the websites and employ any other online resources. We did not choose any commercial tool such as Adobe Illustrator as the baseline, because the learning curve is quite high for general users and the interfaces are dramatically different.

Participants and Apparatus. We recruited 24 participants (10 females; aged 19–26) from a local university. They

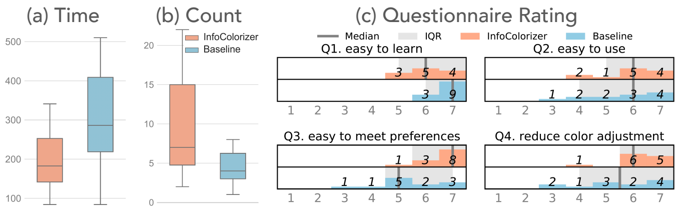

Fig. 8. Results of the controlled user study: (a) Completion time for generating satisfying infographics in Task 1. (b) Counts of the resulting infographics in Task 2. (c) Participants' subjective ratings on the exit-questionnaire (the higher is better).

are all with normal color vision and their backgrounds range from engineering, law, to business. Their average years of experience in visualization or design is 0.375 ($\sigma = 0.77$), so that they are novice users for our study tasks. Their self-reported expertise of color theories (e.g., harmony rules, color semantics) was: $M = 2$ and $IQR = 2$, on a 7-point Likert scale (1="do not know at all" and 7="very familiar") We deployed InfoColorizer and its baseline version on the cloud, and participants completed the study remotely via video conferencing software on their own machines.

Tasks. We created four experimental infographics with certain contextual information (e.g., talking about a kid's weekend). Participants needed to complete two tasks during a study session. Task 1 aimed to assess the efficiency of the tool (S1), in which participants needed to color three infographics (out of the four) until they were satisfied with the results, one by one, without a time limitation. For each infographic, according to the context, three forms of color preferences were specified for three elements during the tasks, including: an exact color, a color name, and a semantic or affective word. Task 2 aimed to assess the creativity supported by the tool (S2), in which participants colored the same infographic (the rest one of the four) within 15 minutes to obtain as many satisfying results as possible. In this task, users were given general contextual information instead of concrete preferences. In each task, we explicitly explained the constraints or context and asked participants to ensure the pre-defined preferences were met and each element was distinguishable. For the baseline, participants could import a color palette as a whole to minimize the effort of copying and pasting single colors from the websites.

Design and Procedure. We employed a between-subjects design, with 12 participants finishing two tasks in each condition. We ensured that each infographic appeared in Task 2 three times in each condition across participants, and counterbalanced the order of the remaining three infographics for Task 1. Each study session began with a tutorial about the tool (i.e., InfoColorizer or Baseline with websites). Then, participants completed a training task on a different infographic (than the four) with similar task requirements. They could ask any questions about the tool. After, they performed Task 1 and then Task 2 in order. In the end, they filled in an exit-questionnaire and the Creativity Support Index questionnaire [13], followed by a semi-structured interview. For participants in the Baseline condition, we also briefly demonstrated InfoColorizer and asked for their comments. Each study session lasted around 1.5 hours and each participant received $12 in compensation.

6.2.2 Results and Analysis: Task Performance

Fig. 8-a and Fig. 8-b show the completion time of Task 1 and the resulting infographics count in Task 2, respectively. An independent-samples t-test showed significant differences on both the completion time ($t = -4.83, p \ll 0.001$) and counts ($t = 2.29, p = 0.02$). This indicates that participants generated satisfying palettes faster with InfoColorizer ($\mu = 197s$, $95\%\ CI = [173, 222]$) than with Baseline ($\mu = 306s$, $95\%\ CI = [265, 346]$) for Task 1. Moreover, for Task 2, participants created more satisfying infographics with InfoColorizer ($\mu = 9.25$, $95\%\ CI = [5.7, 12.8]$) than with Baseline ($\mu = 4.7$, $95\%\ CI = [3.4, 5.9]$).

6.2.3 Results and Analysis: Participants' Feedback

To further answer S1, we analyzed the ratings on the exit-questionnaire (Fig. 8-c; 7-point Likert scale with 1="strongly disagree" and 7="strongly agree"). The questions include: the tool is easy to use (Q1); the tool is easy to learn (Q2); with the tool, it is easy to meet color preferences for certain elements (Q3); the tool reduces the burden for color adjustment (Q4). We performed a Mann-Whitney test on each question.

For Q1, participants generally thought Baseline was easier to learn ($U = 37.5, p = 0.014$), which might be because the interface of InfoColorizer was more complicated with the recommendation function. Nevertheless, all participants gave a positive rating (≥ 5) for InfoColorizer.

There was no significance on Q2 ($U = 63.5, p = 0.32$), but InfoColorizer in general received more positive ratings. This could be because it is equipped with the recommendation that benefited novices by *"reducing the search space"* (P6). After being demonstrated InfoColorizer, P1 using the Baseline said: *"It is exactly what I want, as it can alleviate my burden to collect, assign and adjust colors."*

A significant difference was found ($U = 31.5, p = 0.007$) on Q3, indicating that InfoColorizer allows participants to easily colorize infographics under specific color constraints. Among the participants using the Baseline, P10 complained: *"Sometimes I cannot get proper colors meeting preferences while looking harmonious from the three websites, so I need to determine colors in a trial-and-error process myself."* Additionally, P12 said that *"I use ColorBrewer in my daily life, but the choices are limited. I also feel inconvenient when I want to change a single color in a certain palette, because the remaining colors cannot be updated."* Participants from the InfoColorizer condition reported that they also encountered such difficulties; however, they felt that InfoColorizer *"integrates knowledge about colors, semantics and emotions,"* (P7) and *"the recommendation can solve these difficulties."* (P8).

Though no significance was found ($U = 49.5, p = 0.091$) on Q4, InfoColorizer ($\mu = 6.25, \sigma = 0.87$) had a higher average score than Baseline ($\mu = 5.42, \sigma = 1.51$) and a smaller variance on Q4, indicating its stability in reducing the burden for color adjustment. P17 using Baseline said: *"I had no idea whether a palette would work well for an infographic before being applied, even if it looked fine on the websites. Thus, I needed to keep trying based on my intuition."* This was also echoed by Baseline users P1 and P12.

6.2.4 Results and Analysis: Creativity Support

To answer S2, we utilized the Creativity Support Index (CSI) [13], which quantifies how well a tool can support

TABLE 1
Participants' scores on the five factors and the final CSI scores (the higher the better) of InfoColorizer (I) and Baseline (B), with independent-samples t-test results.

CSI Dimension	I	B	T-test
Expressiveness	33.8	26.7	$t = 2.84, p = 0.009$
Exploration	35.2	26.5	$t = 3.38, p = 0.003$
Results Worth Effort	36.3	25.8	$t = 9.71, p \ll 0.001$
Immersion	32.5	27.2	$t = 2.08, p = 0.049$
Enjoyment	35.8	29	$t = 3.85, p = 0.001$
CSI	57.9	45.1	$t = 4.75, p = 0.0001$

users' creativity based on a research-tested questionnaire. One usage of CSI is to compare two tools used by people from two groups for the same tasks, which is well-suited to our study. Following the practices in the previous work [52], [53], we asked users to rate the corresponding study system on five factors: Expressiveness, Exploration, Results Worth Effort, Immersion, and Enjoyment. Table 1 shows each factor's scores and the calculated final CSI scores.

Overall, InfoColorizer received a mean CSI score of 57.9 ($\sigma = 6.51$), much better than that of Baseline: a mean of 45.1 ($\sigma = 6.72$). A independent-samples t-test showed a significant difference ($t = 4.75, p = 0.0001$), indicating InfoColorizer performed considerably better than Baseline. Moreover, InfoColorizer significantly outperformed Baseline on all the factors. The biggest difference existed in Results Worth Effort. Though some recommended palettes were inferior to their expectation, participants still thought the overall recommendations were acceptable and the palette refinement workflow of InfoColorizer was smoother than Baseline. Thus, they could obtain qualified and satisfactory palettes for an infographic with less effort. The experience in Task 2 influenced how participants felt about the Expressiveness and Exploration of InfoColorizer and Baseline, because they needed to be creative to provide as many solutions as possible. Several participants reported that InfoColorizer sometimes surprised them by recommending palettes of different styles that they had never thought about, so that they could explore many different possibilities. On the contrary, P7 using Baseline said that *"The websites can help me search colors when I have an idea, but they cannot provide me with ideas."* To further improve Expressiveness, P16 and P17 suggested including more vague descriptive words into InfoColorizer. The better performance on Immersion and Enjoyment of InfoColorizer may be related to the fact that it has a higher degree of integration than Baseline. This was because participants could find colors under specific preferences and assign them to elements within a single system instead of several websites.

6.2.5 Results and Analysis: Participants' Behaviors

We qualitatively investigated participants' behavioral patterns based on our observation, revealing that they adopted different ways to get started, obtain the first palettes, modify colors, and adjust color assignment.

Baseline. For Task 1, it was rare for participants to use a complete palette directly from the online sources [6], [2], [1], since none could meet all the color preferences. Most participants started with elements having the constraints. We noticed that Coolers [2] was more popular for them to get started. The reasons might be: 1) compared to ColorBrewer [1], Coolers support searching palettes with keywords; and 2) Coolers has more flexible interactions to generate palettes with gradient and with over five colors than Adobe Color [6]. After coloring these elements, participants usually chose colors from the color picker and swatch embedded in the system for the remaining elements. The swatch consisted of palettes pre-loaded from ColorBrewer and imported by them from Coolor and Adobe Color previously. The adjustment of color palettes and color assignment happened in various ways, including: 1) participants might adjust colors both before and after obtaining the first complete palettes; 2) some finished quickly only via one to two adjustments, while others repeatedly assigned several colors to one element and compared the corresponding results; and 3) they usually either focused on elements in an arbitrary order or just from left to right, and they tended to adjust elements locally if the elements were clustered visually.

For Task 2 without requirements, they relied more on complete palettes in the provided tools by changing one to three colors or just trying different color assignments. When searching palettes in Adobe Color and Coolors, some participants only focused on whether palettes were aesthetic while others might filter palettes using keywords fitting the infographic topic. In both tasks, we observed that all participants did not use other online coloring tools, nor did they use the color wheel and harmony rules provided by Adobe Color. This might be because they were novice users and unfamiliar with other tools and the rules.

InfoColorizer. Users behaved much more consistently than Baseline on the two tasks. They started with specifying preferences by inputting words, getting recommendations, and bookmarked palettes they liked. They might adjust one to two unsatisfactory colors by using the color picker, swatch, or recommendation functions. More participants requested recommendations again since it allowed them to obtain many possible results. They often finished one infographic in Task 1 within two requests. As for Task 2, they usually got the first satisfactory result within two requests and had bookmarked two to four palettes, from which they could derive more palettes in the next request. We observed that they had less hesitation and pauses during the creation than participants using Baseline. This might be because that InfoColorizer provided them with a more consistent workflow, reduced their mental effort, and allowed them to focus on points of interest.

6.3 Survey Study

To evaluate the quality of recommended palettes, we conducted a survey study with infographic readers and compared five conditions: artist-designed, InfoColorizer-recommended, Baseline-crafted ColorBrewer-predefined, and randomly-generated palettes.

6.3.1 Study Setup

We used the four infographics mentioned in Sec. 6.2. For each infographic, a professional designer created one palette for artist-designed condition and we obtained nine palettes for each of the other four conditions. Specifically, for InfoColorizer and Baseline conditions, we utilized the palettes

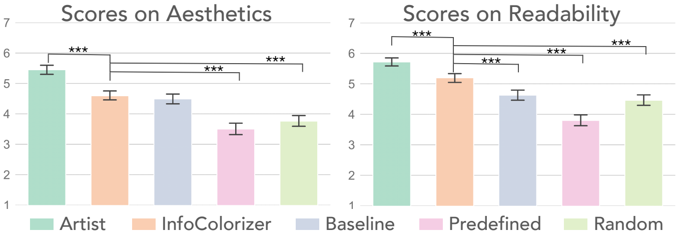

Fig. 9. Survey study results: the average scores on aesthetics and readability with 95% CIs (the higher is better; *** denotes $p < .001$).

crafted in Task 2 of the controlled user study, because no concrete preferences were set in Task 2. Thus, the palettes were produced under the same settings among the artist-designed, InfoColorizer, and Baseline conditions. For the ColorBrewer condition, we only considered categorical palettes and randomly assigned their colors to infographic elements. For randomly-generated condition, we randomly generated palettes and applied them to infographics.

We formulated the study as online surveys. Each survey has four problem sets, each containing five pictures derived from an identical infographic but colored with five palettes, each for one of the five conditions. The artist-designed palettes were repeated across all surveys; for the other four conditions, the palettes were randomly selected from the corresponding generated ones above. Therefore, each survey contained $4 \times 5 = 20$ pictures. For each problem set, we asked participants to provide two 7-point scores (the higher is better), on color aesthetics and element readability. We randomized the order of conditions within each problem set, as well as the order of the problem sets.

6.3.2 *Participants*

We released the survey on Amazon Mechanical Turk, and collected 102 responses, all valid. Their demographic information is: 81 males and 21 females, aged 17–57 ($\mu = 32.3, \sigma = 8.5$), 0–16 years ($\mu = 2.2, \sigma = 3.5$) of experience in visualization or design, and all with normal color vision. They completed the study on their own machines.

6.3.3 *Results and Analysis*

Fig. 9 shows the scores of five conditions on color aesthetics and element readability. We can see that artist-designed palettes received the highest average scores on both color aesthetics ($\mu = 5.45, 95\%\ CI = [5.31, 5.59]$) and readability ($\mu = 5.72, 95\%\ CI = [5.59, 5.85]$), respectively. This is because these palettes were carefully designed by the professional designer. Among the rest, InfoColorizer-recommended palettes obtained the highest overall scores in aesthetics ($\mu = 4.60, 95\%\ CI = [4.44, 4.75]$) and readability ($\mu = 5.20, 95\%\ CI = [5.06, 5.34]$). A Friedman test (used for comparing more than two conditions) indicated significant differences among different methods for aesthetics ($\chi^2 = 261.27, p < 0.001$) and readability ($\chi^2 = 267.6, p < 0.001$). Post-hoc Dunn tests with Bonferroni adjustment (used for pairwise comparison) showed that the score of InfoColorizer-recommended palettes was significantly higher than that of randomly-generated palettes both on aesthetics ($z = 6.64, p < 0.001$) and readability ($z = 6.11, p < 0.001$), as well as significantly higher than that of ColorBrewer-predefined palettes both on aesthetics ($z = 8.53, p < 0.001$) and readability ($z = 10.96, p < 0.001$). It is not surprising that InfoColorizer outperformed the random condition because InfoColorizer had learned good practice embedded in expert-designed infographics while random palettes were sampled from the whole color space. Before study, we thought that ColorBrewer might be better than InfoColorizer on aesthetics but worse on readability. However, the survey results show that InfoColorizer outperformed in both aspects, indicating that even a good predefined palette can have poor aesthetics and readability because of improper color assignment. While there was no significance between InfoColorizer-recommended and baseline-crafted palettes on aesthetics ($z = 0.69, p = 0.49$), the average score of InfoColorizer was significantly higher readability than that of Baseline ($z = 4.75, p < 0.001$). This indicates that InfoColorizer not only improved users' creation efficiency as shown in Sec. 6.2, but also ensured that the recommended palettes had at least the same level of quality as user-crafted palettes with the Baseline on both aesthetics and readability.

TABLE 2
Participants' ratings in the interview study.

	P1	P2	P3	P4
Q1 Tool is easy to learn	6	6	6	6
Q2 Tool is easy to use	7	6	4	5
Q3 Infographics element layers panel is intuitive	6	7	6	6
Q4 Specifying color preferences is useful	7	7	7	7
Q5 Iteratively refining the results is useful	6	7	6	7
Q6 Recommended palettes look good	6	4	5	5
Q7 Tool makes starting to design a palette easier	6	7	6	7
Q8 Tool makes the design process more effective	6	5	6	7

6.4 Interview Study

To investigate how the tool looks from the eyes of experts, we conducted an interview study to collect in-depth qualitative feedback from graphic designers, in which they used InfoColorizer to design palettes for infographics of their own interests, in a more realistic setting.

6.4.1 *Study Setup*

We recruited four experts (two females and two males) with over three years of experience in designing infographics and visualizations from an online community. Each study session started with a 10-minute survey about the tools or methods they used to obtain palettes and the difficulties encountered when using the tools. Then, they explored InfoColorizer and asked questions after a short tutorial (20 minutes). Next, they completed a task (e.g., colorizing an infographic freely) (20 minutes) and filled a questionnaire to rate the system's functionality (10 minutes). We encouraged them to think aloud during the study. An experimenter was responsible for answering questions and taking notes.

6.4.2 *Results and Analysis*

All participants could complete the task and their ratings are shown in Table 2. Specifically,

Ease of Learning and Use (Q1, Q2). All participants found InfoColorizer very easy to learn, as P1 explained *"the* *tool is designed similar to tools I regularly use... layers and panels are similar to document or photo editors."* The responses for ease of use were mixed. P2 thought *"the functions are pretty intuitive"* but P3 commented *"some buttons are new to me... I do not know what they do without explanations."*

Infographic Elements Visualization (Q3). They found the visualization of layered elements (Fig. 6-C1) very intuitive. For example, P1 commented *"this looks familiar to me, just like the layers panel in Photoshop"* and P2 gave similar feedback that *"if you are an Adobe user, you will understand this immediately."* One limitation was that the visualization is *"missing the vertical spacial order"*, as P1 explained *"sometimes it is hard to locate elements at the same horizontal location."* P1 and P2 also suggested supporting row selections, as P2 said *"I hope to select layer by layer... I usually select or edit elements in a layer to bind or assign them the same color."*

Color Preference Setting (Q4). They gave the highest ratings for the Color Preferences panel (Fig. 6-C1). Specifically, P1 liked the combination of manual (i.e., color assignment) and automatic (i.e., vague preference) methods and said *"they are very complementary to each other... some are manual but can see effects immediately... some are automatic but need to re-run the model."* Besides, P4 particularly liked the color binding feature and explained *"it allows users to input the relationships among the elements to the model... the model won't be able to know this information without human input."* P2 suggested that *"it will be amazing if it can learn and memorize my color preferences from my design history."*

Recommendation Quality (Q6). Most participants thought the quality of the recommended palettes was good, as P1 applauded that *"the color contrast between the background and foreground is good."* Two participants suggested grouping the recommendations by similarity, as P4 explained that *"I hope the model can group similar palettes together, so I only need to try one from each group to quickly find out which style is the best."* In addition, P3 asked for more explanations behind each recommendation: *"suppose the system is already learning from professional designers, maybe it can explain the recommendation by telling me the styles or names of the designers, so I can learn their styles. Great for non-expert designers to improve their skills by learning."*

Iterative Workflow (Q5, Q7). They thought InfoColorizer made it easy to get started and that the iterative refinement process was effective for producing high-quality designs. For example, P2 commented that *"usually I don't know what color I want at the beginning... so having some recommendations is helpful."* P1 found the iterative refinement process very effective and explained that *"the initial recommendation already looks good but a little bit diverse... after I specify preferences, it narrows down the design space and starts to give more personalized recommendations."* Similarly, P4 also reflected on her design iterations and commented that *"at the beginning, the search space is usually large... with the recommendations, I only need to review a few to identify what styles fit the best, so I can quickly reduce the search space."* To improve the design process, P1 hoped InfoColorizer *"can memorize my history, so the system will become smarter even at the first recommendation."* P3 pointed out a limitation that *"the recommendations tend to be similar to what I selected in the last round, even though I did not want to."*

Comparison to Existing Tools (Q8). The tools they regularly use fall into three categories: 1) manual, such as assigning colors in Photoshop or Illustrator, and 2) half-manual, such as tools recommending high contrast colors to a specified color, and 3) templates, such as themes in PowerPoint. Compared to existing tools, most participants thought InfoColorizer optimizes the palette design process since it is more automatic and personalized. Compared to the template tools, P2 commented that *"it (InfoColorizer) is more flexible than pre-defined themes... I can specify color preferences and choose from a large set of options"* and P3 added that *"pre-defined themes are not considering the specific design I want to make... it always give you the same set of templates and makes your design look similar to others."* Compared to the manual or half-manual tools, P4 thought InfoColorizer is more efficient since *"manual is not scalable... I can process more designs with the recommendation workflow."* All participants mentioned that InfoColorizer can be used along with their existing tools. P1 explained in detail that *"Here (InfoColorizer) we are starting from scratch. I hope to start from some pre-defined themes... It is more a trade-off between quality and efficiency for different use cases and scenarios."* These comments echo the three challenges of existing tools that users are facing, as described earlier in Sec. 1.

7 Discussion

Here we discuss several aspects about InfoColorizer and our studies, including limitations and potential solutions.

Generalization for Different Infographics. Our current recommendation engine is designed for infographics that can be described by the conceptual model in Sec. 4.2. However, there are some percentage of infographics containing data charts (e.g., line charts, scatterplots). We observed about 1% of such infographics in InfoVIF. While not a big percentage, InfoColorizer may fail to recommend proper palettes for these infographics, because embedded data is not characterized in input features. This can be addressed by collaborating with tools for data charts [22], [19]. For example, to colorize an infographic with a bar chart, a user can first obtain colors from Palettailor [35], assign them to each bar as color preferences, and get recommended colors for the rest elements with InfoColorizer. This process may be further automated by integrating prior work on colorizing data charts [62], [30] as the conditions of VAEAC.

InfoColorizer specifically focuses on recommending color pallettes for infographics that have a very different content and structure compared to natural images. To achieve our goal, we employ a conceptual model (Sec. 4.2) together with extracting various infographic-specific features (Sec. 5.1). The model contains both artistic and graphical data elements and describes an infographic from both logical and spatial aspects, which also enables us to characterize an infographic with three levels of non-color features. Thus, with the power of VAEAC, we are able to capture high-level design practices from a large collection of infographics, and thus recommend suitable color palettes. In-depth comparision of InfoColorizer with colorization tools for natural images is left for future work.

Robustness. As the visual design of infographics is diverse, the rule-based feature extraction algorithm (Sec. 5.1.1) may fail to obtain correct features. The failing cases mainly fall into two categories. First, the method cannot understand complex clipart images, which often serve as semantic backgrounds and may affect the color appearance of the entire infographic. For example, it treats a rocket as several separate shapes rather than as a whole. However, with advanced computer vision techniques, semantic objects can be integrated into our tree model, and thus we can still apply our data-driven method to learn design practices and recommend palettes. Second, the tree construction and recommended palettes will be influenced (Sec. 5.2) if the bounding boxes of some elements are detected wrongly. One solution to address this problem is to allow users to correct tree structures with the interface (Fig. 6-C1).

Explicit and Implicit Color Constraints. When recommending palettes, we only consider color preferences explicitly assigned by users, and leave aside possible implicit constraints exhibited in infographics. For example, if a sequential palette is used for encoding data in an infographic, our generated palettes may fail to remain the relationship. Currently, a user has to specify the sequential relationships using exact colors or vague words (e.g., Fig. 7-B3). A solution can be to embed such relative color relations (e.g., sequential) into the feature vector, and train VAEAC to learn these patterns. Similarly, as infographics can benefit from using gradient colors, integrating gradient colors into the features would be interesting to explore.

Generalization and Penalization Trade-off. Our method is data-driven, meaning that the style and quality of recommended palettes depend on the training data. More training data will likely enhance the model with generality and accuracy. However, whether a palette is aesthetic or not is still subjective. Currently, we choose VAEAC which can generate diverse, as well as relevant, palettes (Appendix Sec. B) to accommodate users with different aesthetic tastes. One solution can be training a more personalized model gradually based on the resulting palettes chosen by a user.

Considerations of UI design. We considered two ways to display recommended results: palettes and infographic previews. The former is more space-efficient but may be less intuitive; there exists a trade-off. By mimicking Adobe Illustrator, we chose to display the results as a list of palettes and use layered rectangles to indicate the tree structure. We did not receive any complaints about this design, as it is used in many commercial infographics authoring tools. The user can still preview the colored infographic by selecting a palette. However, further studies are required to assess these two designs. Also, our user interface can be further improved. For example, integrating the operation panels to the main canvas may streamline the whole workflow.

Limitations in Study Design. For the survey study, we randomly assigned colors to infographic elements for the ColorBrewer condition. Without manual adjustment, adjacent elements may be assigned with the same color, reducing the readability. However, this is a common situation in real world; and one of the challenges that we address here is the color assignment problem. Further, we note that the sample size of our interview study might be small. However, we obtained deeper insights regarding InfoColorizer and their infographic creation workflow, and our controlled study with more users complements this effect to some extent. Regardless, a future deployment study may be needed to evaluate the usefulness of InfoColorizer with more realistic settings for the long term.

8 Conclusion and Future Work

We have introduced InfoColorizer, an interactive system that supports effective infographic color palette design via cogent recommendations. The system leverages a conceptual infographic model and deep learning techniques to lower design barriers, support flexible color preference specification, and adapt palette generation to spatial relationships of infographic elements. We have demonstrated the effectiveness and usefulness of InfoColorizer through case studies, a controlled user study, a survey study, and an interview study. Our work opens several avenues for future work. We plan to explore metrics to rank returned palette recommendations, which can further reduce users' effort to examine and choose palettes. We also would like to support more advanced color preferences such as relative lightness and perceptual differences between two elements.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments. This work is supported in part by the NSERC Discovery Grant, a research grant from Microsoft Research Asia, and an Adobe gift fund.

References

[1] Color brewer. https://colorbrewer2.org/, Accessed in 2020.

[2] Coolors. https://coolors.co/, Accessed in 2020.

[3] MassVis. http://massvis.mit.edu/, Accessed in 2020.

[4] SVG Infographics. https://all-free-download.com/free-vector/svg-infographic.html, Accessed in 2020.

[5] Viusally29K. http://visdata.mit.edu/, Accessed in 2020.

[6] Adobe Color. https://color.adobe.com/, Accessed in 2020.

[7] L. Bartram, A. Patra, and M. Stone. Affective color in visualization. In *Proc. ACM CHI*, pages 1364–1374, 2017.

[8] S. Bateman, R. L. Mandryk, C. Gutwin, A. Genest, D. McDine, and C. Brooks. Useful junk? the effects of visual embellishment on comprehension and memorability of charts. In *Proc. ACM CHI*, pages 2573–2582, 2010.

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

[10] M. Bostock, V. Ogievetsky, and J. Heer. D^3 data-driven documents. *IEEE TVCG*, 17(12):2301–2309, 2011.

[11] Z. Bylinskii, N. W. Kim, P. O'Donovan, S. Alsheikh, S. Madan, H. Pfister, F. Durand, B. Russell, and A. Hertzmann. Learning visual importance for graphic designs and data visualizations. In *Proc. ACM UIST*, pages 57–69, 2017.

[12] Z. Chen, Y. Wang, Q. Wang, Y. Wang, and H. Qu. Towards automated infographic design: Deep learning-based auto-extraction of extensible timeline. *IEEE TVCG*, 26(1):917–926, 2019.

[13] E. Cherry and C. Latulipe. Quantifying the creativity support of digital tools through the creativity support index. *ACM TOCHI*, 21:1–25, 2014.

[14] D. Cohen-Or, O. Sorkine, R. Gal, T. Leyvand, and Y.-Q. Xu. Color harmonization. In *Proc. ACM SIGGRAPH*, pages 624–630. 2006.

[15] W. Cui, X. Zhang, Y. Wang, H. Huang, B. Chen, L. Fang, H. Zhang, J.-G. Lou, and D. Zhang. Text-to-viz: Automatic generation of infographics from proportion-related natural language statements. *IEEE TVCG*, 26(1):906–916, 2019.

[16] A. Desolneux, L. Moisan, and J.-M. Morel. Gestalt theory and computer vision. In *Seeing, Thinking and Knowing*. 2004.

[17] V. Dibia and Ç. Demiralp. Data2vis: Automatic generation of data visualizations using sequence to sequence recurrent neural networks. *IEEE CG&A*, 39(5):33–46, 2018.

[18] E. Feichtner and I. MÃžller. On the topology of nested set complexes. *Proc. of the American Mathematical Society*, 133(4):999–1006, 2005.

[19] C. C. Gramazio, D. H. Laidlaw, and K. B. Schloss. Colorgorical: Creating discriminable and preferable color palettes for information visualization. *IEEE TVCG*, 23(1):521–530, 2016.

[20] S. Haroz, R. Kosara, and S. L. Franconeri. Isotype visualization: Working memory, performance, and engagement with pictographs. In *Proc. ACM CHI*, pages 1191–1200, 2015.

[21] L. Harrison, K. Reinecke, and R. Chang. Infographic aesthetics: Designing for the first impression. In *Proc. ACM CHI*, pages 1187–1190, 2015.

[22] M. Harrower and C. A. Brewer. Colorbrewer. org: an online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003.

[23] J. Heer and M. Stone. Color naming models for color selection, image editing and palette design. In *Proc. ACM CHI*, 2012.

[24] K. Hu, M. A. Bakker, S. Li, T. Kraska, and C. Hidalgo. Vizml: A machine learning approach to visualization recommendation. In *Proc. ACM CHI*, pages 1–12, 2019.

[25] O. Ivanov, M. Figurnov, and D. Vetrov. Variational autoencoder with arbitrary conditioning. In *Proc. ICLR*, 2019.

[26] G. Jalal, N. Maudet, and W. E. Mackay. Color portraits: From color picking to interacting with color. In *Proc. ACM CHI*, pages 4207–4216, 2015.

[27] J. Kim and C. D. Scott. Robust kernel density estimation. *Journal of Machine Learning Research*, 13(Sep):2529–2565, 2012.

[28] N. W. Kim, E. Schweickart, Z. Liu, M. Dontcheva, W. Li, J. Popovic, and H. Pfister. Data-driven guides: Supporting expressive design for information graphics. *IEEE TVCG*, 23(1), 2016.

[29] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *Proc. ICLR*, 2014.

[30] S. Lee, M. Sips, and H.-P. Seidel. Perceptually driven visibility optimization for categorical data visualization. *IEEE TVCG*, 19(10):1746–1757, 2012.

[31] H. Levkowitz and G. T. Herman. Color scales for image data. *IEEE CG&A*, (1):72–80, 1992.

[32] S. Lin, J. Fortuna, C. Kulkarni, M. Stone, and J. Heer. Selecting semantically-resonant colors for data visualization. *Comput. Graph. Forum*, 32(3):401–410, 2013.

[33] S. Lin, D. Ritchie, M. Fisher, and P. Hanrahan. Probabilistic color-by-numbers: Suggesting pattern colorizations using factor graphs. *ACM TOG*, 32(4):1–12, 2013.

[34] Z. Liu, J. Thompson, A. Wilson, M. Dontcheva, J. Delorey, S. Grigg, B. Kerr, and J. Stasko. Data illustrator: Augmenting vector design tools with lazy data binding for expressive visualization authoring. In *Proc. ACM CHI*, pages 1–13, 2018.

[35] K. Lu, M. Feng, X. Chen, M. Sedlmair, O. Deussen, D. Lischinski, Z. Cheng, and Y. Wang. Palettailor: Discriminable colorization for categorical data. *IEEE TVCG*, 27(2):475–484, 2021.

[36] M. Lu, S. Wang, J. Lanir, N. Fish, Y. Yue, D. Cohen-Or, and H. Huang. Exploring visual information flows in infographics. In *Proc. ACM CHI*, pages 770–779, 2020.

[37] Y. Luo, X. Qin, N. Tang, and G. Li. DeepEye: Towards automatic data visualization. In *Proc. IEEE ICDE*, pages 101–112, 2018.

[38] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM TOG*, 5(2), 1986.

[39] B. J. Meier, A. M. Spalter, and D. B. Karelitz. Interactive color palette tools. *IEEE CG&A*, 24(3):64–72, 2004.

[40] S. Mittelstädt, A. Stoffel, and D. A. Keim. Methods for compensating contrast effects in information visualization. *Comput. Graph. Forum*, 33(3):231–240, 2014.

[41] K. Moreland. Diverging color maps for scientific visualization. In *Intl. Sym. on Visual Computing*, pages 92–103, 2009.

[42] D. Moritz, C. Wang, G. L. Nelson, H. Lin, A. M. Smith, B. Howe, and J. Heer. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *IEEE TVCG*, 25(1):438–448, 2019.

[43] P. O'Donovan, A. Agarwala, and A. Hertzmann. Color compatibility from large datasets. In *Proc. ACM SIGGRAPH*, 2011.

[44] S. E. Palmer, K. B. Schloss, and J. Sammartino. Visual aesthetics and human preference. *Annual review of psychology*, 64:77–107, 2013.

[45] Z. Qu and J. Hullman. Keeping multiple views consistent: Constraints, validations, and exceptions in visualization authoring. *IEEE TVCG*, 24(1):468–477, 2018.

[46] Ramer-Douglas-Peucker. Ramer-douglas-peucker algorithm. https://karthaus.nl/rdp/, 1972.

[47] S. F. Roth, J. Kolojejchick, J. Mattis, and J. Goldstein. Interactive graphic design using automatic presentation knowledge. In *Proc. ACM CHI*, pages 112–117, 1994.

[48] B. Saleh, M. Dontcheva, A. Hertzmann, and Z. Liu. Learning style similarity for searching infographics. *arXiv preprint arXiv:1505.01214*, 2015.

[49] A. Satyanarayan and J. Heer. Lyra: An interactive visualization design environment. *Comput. Graph. Forum*, 33(3):351–360, 2014.

[50] V. Setlur and M. C. Stone. A linguistic approach to categorical color assignment for data visualization. *IEEE TVCG*, 22(1):698–707, 2015.

[51] X. Shu, A. Wu, J. Tang, B. Bach, Y. Wu, and H. Qu. What makes a data-gif understandable? *IEEE TVCG*, 27(2):1492–1502, 2021.

[52] M. Shugrina, J. Lu, and S. Diverdi. Playful palette: an interactive parametric color mixer for artists. *ACM TOG*, 36, 2017.

[53] M. Shugrina, W. Zhang, F. Chevalier, S. Fidler, and K. Singh. Color builder: A direct manipulation interface for versatile color theme authoring. In *Proc. ACM CHI*, pages 1–12, 2019.

[54] S. Smart, K. Wu, and D. A. Szafir. Color crafting: Automating the construction of designer quality color ramps. *IEEE TVCG*, 26(1):1215–1225, 2019.

[55] S. Suzuki. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985.

[56] D. A. Szafir. Modeling color difference for visualization design. *IEEE TVCG*, 24(1):392–401, 2018.

[57] T. Tang, R. Li, X. Wu, S. Liu, J. Knittel, S. Koch, L. Yu, P. Ren, T. Ertl, and Y. Wu. Plotthread: Creating expressive storyline visualizations using reinforcement learning. *IEEE TVCG*, 27(2):294–303, 2020.

[58] M. Tennekes and E. de Jonge. Tree colors: color schemes for tree-structured data. *IEEE TVCG*, 20(12):2072–2081, 2014.

[59] C. Tominski, G. Fuchs, and H. Schumann. Task-driven color coding. In *Proc. Intl. Conf. Info. Vis.*, pages 373–380, 2008.

[60] A. Tremeau and N. Borel. A region growing and merging algorithm to color segmentation. *Pattern recognition*, 30(7):1191–1203, 1997.

[61] S. van Buuren and K. Groothuis-Oudshoorn. Mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software, Articles*, 45(3):1–67, 2011.

[62] Y. Wang, X. Chen, T. Ge, C. Bao, M. Sedlmair, C.-W. Fu, O. Deussen, and B. Chen. Optimizing color assignment for perception of class separability in multiclass scatterplots. *IEEE TVCG*, 25(1):820–829, 2018.

[63] Y. Wang, Z. Sun, H. Zhang, W. Cui, K. Xu, X. Ma, and D. Zhang. Datashot: Automatic generation of fact sheets from tabular data. *IEEE TVCG*, 26(1):895–905, 2019.

[64] Y. Wang, H. Zhang, H. Huang, X. Chen, Q. Yin, Z. Hou, D. Zhang, Q. Luo, and H. Qu. Infonice: Easy creation of information graphics. In *Proc. ACM CHI*, pages 1–12, 2018.

[65] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Towards a general-purpose query language for visualization recommendation. In *Proc. of the Workshop on Human-In-the-Loop Data Analytics*, pages 1–6, 2016.

[66] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE TVCG*, 22(1):649–658, 2016.

[67] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proc. ACM CHI*, pages 2648–2659, 2017.

[68] A. Wu, L. Xie, B. Lee, Y. Wang, W. Cui, and H. Qu. Learning to automate chart layout configurations using crowdsourced paired comparison. In *Proc. ACM CHI*, 2021.

[69] H. Xia, N. Henry Riche, F. Chevalier, B. De Araujo, and D. Wigdor. Dataink: Direct and creative data-oriented drawing. In *Proc. ACM CHI*, pages 1–13, 2018.

[70] J. Yoon, J. Jordon, and M. van der Schaar. GAIN: Missing data imputation using generative adversarial nets. In *Proc. ACM ICML*, pages 5689–5698, 2018.

[71] N. Zhao, Y. Cao, and R. W. Lau. What characterizes personalities of graphic designs? *ACM TOG*, 37(4):1–15, 2018.

Lin-Ping Yuan is currently a Ph.D. student in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology (HKUST). She obtained her B.Eng. degree in Software Engineering from Xi'an Jiaotong University, China in 2019. Her research interests include information visualization, visual analytics, and VR/AR.

Ziqi Zhou is currently an M.Math. student in the Cheriton School of Computer Science at the University of Waterloo. His research interests include information visualization, human-computer interaction, and data science.

Jian Zhao is an assistant professor in the Cheriton School of Computer Science at the University of Waterloo, where he directs the WatVis (Waterloo Visualization) group. His research interests include information visualization, human-computer interaction, and data science. His work contributes to the development of advanced interactive visualizations that promote the interplay of human, machine, and data.

Yiqiu Guo is currently an undergraduate student in the Department of Software Engineering at Xi'an Jiaotong University (XJTU). His research interests are in Computer Vision and Deep Learning.

Fan Du is a research scientist at Adobe Research. He received his Ph.D. degree in Computer Science from the University of Maryland, College Park. His research focuses on data visualization and human-computer interaction, especially on analyzing electronic health records and user activity logs.

Huamin Qu Huamin Qu is a professor in the Department of Computer Science and Engineering (CSE) at the Hong Kong University of Science and Technology (HKUST) and also the director of the interdisciplinary program office (IPO) of HKUST. He obtained a BS in Mathematics from Xi'an Jiaotong University, China, an MS and a PhD in Computer Science from the Stony Brook University. His main research interests are in visualization and human-computer interaction, with focuses on urban informatics, social network analysis, E-learning, text visualization, and explainable artificial intelligence (XAI).