8-2021

A survey on ML4VIS: Applying machine learning advances to data visualization

Qianwen WANG

Zhutian CHEN

Yong WANG
Singapore Management University, yongwang@smu.edu.sg

Huamin QU

Citation

A Survey on ML4VIS: Applying Machine Learning Advances to Data Visualization

Qianwen Wang, Zhutian Chen, Yong Wang, and Huamin Qu

Abstract—Inspired by the great success of machine learning (ML), researchers have applied ML techniques to visualizations to achieve a better design, development, and evaluation of visualizations. This branch of studies, known as ML4VIS, is gaining increasing research attention in recent years. To successfully adapt ML techniques for visualizations, a structured understanding of the integration of ML4VIS is needed. In this paper, we systematically survey 88 ML4VIS studies, aiming to answer two motivating questions: *"what visualization processes can be assisted by ML?"* and *"how ML techniques can be used to solve visualization problems?"* This survey reveals seven main processes where the employment of ML techniques can benefit visualizations: *Data Processing4VIS, Data-VIS Mapping, Insight Communication, Style Imitation, VIS Interaction, VIS Reading, and User Profiling*. The seven processes are related to existing visualization theoretical models in an ML4VIS pipeline, aiming to illuminate the role of ML-assisted visualization in general visualizations. Meanwhile, the seven processes are mapped into main learning tasks in ML to align the capabilities of ML with the needs in visualization. Current practices and future opportunities of ML4VIS are discussed in the context of the ML4VIS pipeline and the ML-VIS mapping. While more studies are still needed in the area of ML4VIS, we hope this paper can provide a stepping-stone for future exploration.

Index Terms—ML4VIS, Machine Learning, Data Visualization, Survey.

---

1 INTRODUCTION

D ATA visualization (**VIS**), which maps data into visual presentations (*e.g.*, position, color), is a powerful approach to uncover hidden insights and communicate compelling stories in data. However, achieving an effective visualization is typically challenging, requiring a large amount of human effort and having a high reliance on expertise, such as graphic design, user experience design, and data analysis.

Machine Learning (**ML**), on the other hand, provides a practical opportunity to relieve the reliance on experts in visualization. By automatically learning knowledge from data, ML enables task completion without explicit instructions from humans. The application of ML techniques can benefit a variety of visualization-related problems. In this paper, we define visualization-related problems as problems that are related to the process of creating, interacting with, and evaluating visualizations. We refer to studies that *apply ML techniques to solve visualization-related problems using the knowledge extracted from data* as **ML4VIS** in this paper. ML4VIS studies can be traced back to 1986 [89] and have been reignited by the recent advances in ML. A series of ML4VIS studies are emerging, covering a wide range of visualization problems

- Qianwen Wang is with Harvard University.
  E-mail: qianwen_wang@hms.harvard.edu
- Zhutian Chen is with the University of California San Diego.
  E-mail: zhutian@ucsd.edu
- Yong Wang is with Singapore Management University.
  E-mail: yongwang@smu.edu.sg
- Huamin Qu is with Hong Kong University of Science and Technology.
  E-mail: huamin@cse.ust.hk
- Part of this work was done when the first three coauthors were with Hong Kong University of Science and Technology.

(*e.g.*, graph layout [50], visualization assessment [54]) and ML techniques (*e.g.*, variational auto-encoders [49], Mask-RCNN [60]).

While ML4VIS generates a stream of new opportunities, it also poses a series of challenges. First, it is not always clear, among various problems related to visualization, which one can be significantly improved by existing ML techniques and which one still requires a high level of human intervention. Roughly applying ML techniques to unsuitable visualization problems may only impose the drawbacks of ML (*e.g.*, uncertainty, inexplainability) and threaten the validity of a visualization without bringing any benefit. Second, given a visualization-related task, selecting a proper ML technique and employing necessary adaptation are crucial yet challenging. There are a plethora of ML techniques, most of which are exclusively suitable for a certain type of problems. For example, to automatically interpret infographics, ML techniques developed for natural-image-understanding are more helpful and relevant than ML techniques developed for speech recognition. Therefore, the success of ML4VIS hinges on a better understanding of both the visualization and the ML, as well as the integration between the two.

Researchers have contributed a series of frameworks and surveys about the integration between visualization and ML, but they mainly investigate how visualization can assist ML in data analysis [90], [91], [92], [93], [94], [95] rather than how ML can be used to solve visualization-related problems. For example, in a state-of-the-art report, Endert et al. [92] depicted a closer integration between visualization and ML techniques through interactive ML. However, the goal of this integration is still to facilitate data analysis in different application domains (*e.g.*, text analytics, multimedia analytics). Some recent surveys have reviewed studies on automatic visualization creation [96], [97], but these surveys consider both ML-based and non-ML-based methods. More importantly, they only focus on one specific visualization-related problem, *i.e.*, visualization creation. The opportunities for applying ML techniques to various visualization-related problems

(*e.g.*, visualization creation, visualization assessment) have not been fully discussed in previous studies. To the best of our knowledge, this paper is the first published survey on ML4VIS.

In this work, we systematically reviewed the literature in related fields including data visualization, human-computer interaction (HCI), ML, and data mining to investigate how ML techniques are employed and adapted for visualization. Build upon existing visualization theoretical models, we proposed in an ML4VIS pipeline, which reveals seven main processes that can benefit from employing ML techniques. The seven visualization processes (*i.e.*, *Data Processing4VIS, Data-VIS Mapping, Insight Communication, Style Imitation, VIS Interaction, VIS Reading, and User Profiling*) are mapped into the main learning tasks in ML to bridge the needs in visualization with the capabilities of ML. We discuss the current practices and future opportunities of ML4VIS in the context of the ML4VIS pipeline and the ML-VIS mapping. Specifically, our contributions are:

- A codification of 88 ML4VIS studies into **seven processes** in visualization that can benefit from the employment of ML techniques (Section 4);
- An **ML4VIS pipeline** that integrates the seven visualization processes with existing visualization models (Section 5) and an **ML-VIS mapping** between the seven visualization processes and an ML taxonomy (Section 6).
- A set of **research challenges and opportunities** derived from our analysis (Section 7).

We hope this paper can provide a stepping stone for further exploration in the area of ML4VIS. An interactive browser of this survey is available at **https://ml4vis.github.io**.

2 RELATED WORK

This paper mainly relates to two streams of literature: surveys that aim to guide ML-assisted design and theoretical models that summarize the process in visualization.

2.1 Guiding ML-assisted Design

As various ML capabilities (*e.g.*, recommendation, interaction prediction) are integrated into user interfaces, many surveys and reviews have been conducted to guide the application of ML techniques in the design of user interfaces [98], [99], [100].

These surveys on general ML-assisted designs have contributed various guidelines and taxonomies that also inform the ML4VIS studies. Efforts to review these guidelines can be traced back to 1999, when Horvitz [100] reviewed and outlined 12 guidelines for coupling automated services with direct user manipulation. These guidelines were later extended by Amershi et al. [99], who distilled 18 design guidelines for human-AI interaction from over 150 AI-related design recommendations. The majority of these guidelines, such as *"learn from user behaviors"*, *"convey the consequence of user interactions"*, can also be applied to the design of ML-assisted interactive visualizations. Apart from design guidelines, taxonomies can also provide a better understanding of ML techniques through a design perspective. Yang et al. [98] analyzed more than 2000 HCI publications using topic modeling and summarized four channels through which ML advances can provide value to users: self-understanding, contextual awareness, optimization, and utility-capability. The four channels can also help people envision new ways of ML techniques to improve visualization. While these surveys shed some light on the study of

ML4VIS, visualizations have certain unique characteristics that require specified investigations. For example, a precise understanding of and an efficient interaction with data are required in visualization, but they are rarely discussed in the general design of user interfaces.

To guide ML-assisted visualization designs, Saket et al. [101] compared and categorized existing approaches for learning visualization design principles based on the learning method and the input features. The authors then described a research agenda for deriving visualization design principles using ML techniques. Even though this study provides a helpful and insightful agenda, it focuses on the creation of visualization and makes little investigation about user interaction and perception, which are important in interactive visualizations. In this study, we present a survey on ML4VIS that covers various visualization processes.

2.2 Theoretical Models in Visualization

Theoretical models (*e.g.*, workflows, pipelines) have become increasingly important in designing, developing, and evaluating visualizations. A variety of theoretical models have been proposed to depict different processes in visualization [91], [102], [103]. For example, Card [104] presented a visualization workflow to describe the process of creating and interacting with visualizations. Munzner [102] proposed a nested model for the design and validation of visualization.

With the growing usage of ML in visualization, more and more studies have included ML techniques as a component in the theoretical models of visualization. Keim et al. [91] presented the first effort to form a general VA pipeline, where ML is included as a data processing module. Follow-up studies proposed theoretical models tailored for different analysis scenarios and ML techniques [105], [106], [107]. For example, El-Assady et al. [107] extended the general VA pipeline to encompass a new visual analytics paradigm, *i.e.*, speculative execution. Sacha et al. [105] introduced a process model to describe the visual interaction with dimension reduction, a widely-used ML technique. Endert et al. [92] reviewed theoretical models that embed ML techniques into visual analytics and listed opportunities for future research. In these studies, ML is integrated with visualization techniques to facilitate the analysis of complex data. The capabilities of ML in enabling more effective visualizations still remain unclear.

Recently, there is a growing research trend on modeling the role of visualization in the design, development, and evaluation of ML [90], [93], [94], [95], [108]. For example, a Dagstuhl Seminar [90] identified a series of research opportunities in combining ML with visualization. Sacha et al. [108] reinterpreted the traditional VA pipeline and proposed a VIS4ML ontology to encompass ML development workflows into VA workflows. Spinner et al. [109] presented a VA framework for interactive and explainable ML. A research gap exists in summarizing how ML can contribute to visualization.

Contrary to those prior theoretical models, we investigated how ML techniques can be used to benefit the creation, evaluation, and interaction of visualizations. We present an ML4VIS pipeline based on our survey of 88 papers. While more studies are still needed in the field of ML4VIS, we hope this pipeline can help map out the current landscape and inform future opportunities.

3 SURVEY METHODOLOGY

To provide an understanding of the current ML4VIS studies, we conducted an analysis of 88 related papers in the field of VIS, HCI,

TABLE 1
Relevant venues

VIS (visualization)	IEEE VIS (InfoVis, VAST, SciVis), EuroVis, PacificVis, TVCG
HCI (human-computer interaction)	CHI, UIST, IUI
DMM (data mining and management)	KDD, SIGMOD, ICDE, VLDB
ML (machine learning)	CVPR, ECCV, ICML, NeurIPS, ICCV, AAAI, IJCAI, ICLR

Fig. 1. (a) The number of ML4VIS studies published in different fields: VIS, HCI, DMM (data mining and management), ML;(b) The number of ML4VIS studies over time.

TABLE 2
All surveyed papers and their codes.

ID	paper	venue	year
1	Sips et al. [1]	EuroVis	2009
2	Gotz and Wen [2]	IUI	2009
3	Savva et al. [3]	UIST	2011
4	Key et al. [4]	SIGMOD	2012
5	Steichen et al. [5]	IUI	2013
6	Brown et al. [6]	TVCG	2014
7	Lalle et al. [7]	IUI	2014
8	Toker et al. [8]	IUI	2014
9	Sedlmair and Aupetit [9]	CGF	2015
10	Mutlu et al. [10]	TiiS	2016
11	Aupetit and Sedlmair [11]	PVis	2016
12	Siegel et al. [12]	ECCV	2016
13	Kembhavi et al. [13]	ECCV	2016
14	Al-Zaidy et al. [14]	AAAI	2016
15	Pezzotti et al. [15]	TVCG	2016
16	Poco et al. [16]	VIS	2017
17	Kwon et al. [17]	VIS	2017
18	Bylinskii et al. [18]	UIST	2017
19	Saha et al. [19]	IJCAI	2017
20	Kruiger et al. [20]	EuroVis	2017
21	Poco and Heer [21]	EuroVis	2017
22	Jung et al. [22]	CHI	2017
23	Bylinskii et al. [23]	arxiv	2017
24	Al-Zaidy and Giles [24]	AAAI	2017
25	Siddiqui et al. [25]	VLDB	2018
26	Gramazio et al. [26]	VIS	2018
27	Moritz et al. [27]	VIS	2018
28	Berger et al. [28]	VIS	2018
29	Wang et al. [29]	VIS	2018
30	Haehn et al. [30]	VIS	2018
31	Luo et al. [31]	SIGMOD	2018
32	Milo and Somech [32]	KDD	2018
33	Zhou et al. [33]	IJCAI	2018
34	Kahou et al. [34]	ICLR	2018
35	Luo et al. [35]	ICDE	2018
36	Fan and Hauser [36]	EuroVis	2018
37	Chegini et al. [37]	EuroVis	2018
38	Kafle et al. [38]	CVPR	2018
39	Kim et al. [39]	CVPR	2018
40	Battle et al. [40]	CHI	2018
41	Dibia and Demiralp [41]	CGA	2018
42	Haleem et al. [42]	CGA	2018
43	Madan et al. [43]	arxiv	2018
44	Yu and Silva [44]	VIS	2019
45	He et al. [45]	VIS	2019
46	Chen et al. [46]	VIS	2019
47	Han and Wang [47]	VIS	2019
48	Chen et al. [48]	VIS	2019
49	Kwon and Ma [49]	VIS	2019
50	Wang et al. [50]	VIS	2019
51	Han et al. [51]	VIS	2019
52	Wall et al. [52]	VIS	2019
53	Fujiwara et al. [53]	VIS	2019
54	Fu et al. [54]	VIS	2019
55	Porter et al. [55]	VIS	2019
56	Jo and Seo [56]	VIS	2019
57	Ma et al. [57]	VIS	2019
58	Wang et al. [58]	VIS	2019
59	Cui et al. [59]	VIS	2019
60	Chen et al. [60]	VIS	2019
61	Wang et al. [61]	VIS	2019
62	Smart et al. [62]	VIS	2019
63	Huang et al. [63]	VIS	2019
64	Hong et al. [64]	PVis	2019
65	Fan and Hauser [65]	EuroVis	2019
66	Ottley et al. [66]	EuroVis	2019
67	Abbas et al. [67]	EuroVis	2019
68	Kassel and Rohs [68]	EuroVis	2019
69	Hu et al. [69]	CHI	2019
70	Fan and Hauser [70]	CGA	2019
71	Kafle et al. [71]	arxiv	2019
72	Mohammed [72]	VLDB	2020
73	Zhang et al. [73]	VIS	2020
74	Wu et al. [74]	VIS	2020
75	Tang et al. [75]	VIS	2020
76	Qian et al. [76]	VIS	2020
77	Wang et al. [77]	VIS	2020
78	Oppermann et al. [78]	VIS	2020
79	Fosco et al. [79]	UIST	2020
80	Giovannangeli et al. [80]	PacificVis	2020
81	Liu et al. [81]	PacificVis	2020
82	Luo et al. [82]	ICDE	2020
83	Lekschas et al. [83]	EuroVis	2020
84	Zhao et al. [84]	CHI	2020
85	Lai et al. [85]	CHI	2020
86	Kim et al. [86]	CHI	2020
87	Lu et al. [87]	CHI	2020
88	Zhou et al. [88]	arxiv	2020

ML, and data mining & management. Each paper is coded based on the **VIS process** and the **ML task**. Even though automatic analysis methods (*e.g.*, topic modeling) are popular among recent survey papers [98], [105], [110], automatic methods usually only provide a high-level understanding of a large number of papers (*e.g.*, more than 2000 [98]). Manual analysis is still needed to extract details for a deep analysis [105], [110]. Therefore, to ensure a thorough analysis, we conducted a manual analysis for the collected papers.

3.1 Survey Scope

We began our analysis by assembling a corpus of ML4VIS papers. We collected all research articles published between 2016 Jan to 2020 Oct in main journals and conferences in the field of VIS, HCI, ML, and data mining & management by directly accessing the venues, as shown in Table 1. We selected impactful journals and conferences in corresponding fields mainly according to Google Scholar Top Publications [111]. The time range was chosen based on two main considerations: a) this time range has covered the majority of the state-of-art ML4VIS studies and is manageable to conduct a detailed manual analysis; b) given that ML techniques are progressing rapidly, the latest research can provide better guidance for follow-up studies. Moreover, papers published before this time range will be included later by going through the representative references of the collected papers. Following the practice in [93], we mainly checked the title and abstract of each paper to strike a balance between efficiency and accuracy. We went through the full manuscript only when the title and abstract cannot provide clear information. During this process, we paid special attention to a set of ML-related keywords (*e.g.*, *"learning"*, *"machine"*, *"training"*, *"AI"*, *"automation"*, *"CNN"*, *"LSTM"*) and visualization-related keywords (*e.g.*, *"visualization"*, *"infographic"*, *"diagrams"*, *"charts"*). After such a round of paper selection and filtering, we obtained 259 papers.

We further carefully checked the 259 papers using the following criteria. First, since this survey aims to understand the current

practices of employing ML techniques for visualizations, we focused on papers that contribute novel techniques and applications. Theory, evaluation, and dataset papers were excluded. Then, we validated the three terms, **ML**, **VIS**, and **4 (for)**, respectively. To validate the term **ML**, a paper should employ ML techniques to learn from training data rather than totally depend on human defined rules. This criterion requires that a training or optimization process is conducted on a collection of sample data. For example, Voyager [112] was excluded from the corpus because it recommends charts using a list of permitted channels and marks defined by human experts. To validate the term **VIS**, ML should be used for visualization-related problems, including the creation, interaction, and evaluation of visualizations. In other words, we consider information visualization, scientific visualization, and visual analytics except those studies that purely use ML techniques for data processing. This criterion distinguishes ML4VIS studies from studies that purely use ML to analyze data. For example, NNVA [113] employs ML techniques to analyze the parameter sensitivity of yeast cell polarization simulations. In this study, visualizations are merely used to represent the sensitivity analysis that are generated by ML. Therefore, NNVA is not an ML4VIS study and was eliminated from the corpus. To validate the term **4 (for)**, a paper should contribute a novel application or an improvement of existing ML techniques. This criterion helps focus on insightful applications and better scope the range of this survey. For example, a large number of papers that directly use the existing dimension reduction methods to visualize high-dimensional data were not included since they fail to improve these techniques or apply these techniques to a new problem. But the paper [20] that employs t-SNE to layout graph and the paper [29] that improves LDA to maximize the perceived separation of humans were included. Following the practice in other surveys [93], [114], we conducted one-round reference-based search and further went through the related work of these papers. To focus on the latest development in ML4VIS and provide insights for follow-up studies, we did not include papers that were published before 2000. We collected 12 referenced papers that satisfy the aforementioned criteria but appeared earlier than 2016 or on other venues. Our final corpus included 88 relevant papers. A summary of the survey statistics is shown in Fig. 1.

3.2 Paper Coding

The paper analysis consisted of three phases. In the first phase, we extracted a brief description for each paper, including the targeted visualization problems, the employed ML models, and the collected training data. In the second phase, we coded the collected papers from a visualization perspective and analyzed what visualization problems are solved by ML techniques. We categorized these visualization problems based on the visualization processes that these problems are related to. We aimed to answer the question **"what visualization processes can be automated by ML?"** from this encoding. We referred to the visualization processes in existing visualization pipelines [103], [104] and modified them accordingly to better fit the context of ML4VIS. Each of the three authors independently coded the content of 60 papers to ensure that each paper was coded by at least two authors. Note that, at this stage, we did not have unified names for all the visualization processes. The three authors then discussed their own codes about visualization processes with all authors. During weekly discussion, we iteratively adapted, split, and refined the
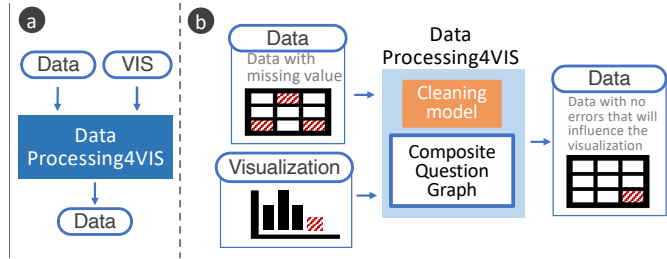

Fig. 2. (a) An illustration of the *Data Processing4VIS*; (b) An example of employing ML in *Data Processing4VIS*: Luo et al. [82]. An ML model (orange box) is trained to detect data errors and propose cleaning options.

codes multiple times until there was no further disagreement and finally arrived at seven main visualization processes. More details of the seven visualization processes are explained in Section 4. Meanwhile, the seven processes form an ML4VIS pipeline, whose differences from existing visualization pipelines are discussed in Section 5. In the third phase, we sought to characterize **"how ML can be used to solve visualization problems?"** As new ML models constantly emerge and one ML model is usually applied to solve various problems, we found that it is not practical to code ML4VIS studies based on the used ML models. Therefore, we answered this question by coding the types of learning tasks. As with phase two, each of the three authors coded 60 papers to ensure each paper was at least coded by two authors. We employed a bottom-up approach and referred to the learning tasks in [115], [116]. The learning tasks are summarized and discussed in Section 6. The codes of all collected papers are summarized in Table 2. For those papers that appeared both at IEEE VIS conference and IEEE TVCG, we listed their venue as VIS. Most of the ML4VIS papers used one ML technique to solve one major visualization problem. But it is also possible that an ML4VIS paper involves more than one learning task and one visualization process, so a few papers are labeled with more than one ML or VIS code.

4 BENEFIT FROM ML: SEVEN VIS PROCESSES

In this section, we introduce seven visualization processes emerged from our literature review that are benefiting from ML techniques, as shown in Table 3. We first explain the transition from inputs to outputs at each visualization process, and then discuss the problems solved by ML at each process with representative examples.

4.1 Data Processing4VIS

Process Description: In *Data Processing4VIS* (Fig. 2(a)), raw data is transformed into a format that better suits the following visualization processes. Contrary to the general-purpose data processing, *Data Processing4VIS* refers to a process that is tightly related to and exclusively designed for the context of visualization, such as enabling efficient visualization creation and enhancing human perception.

Problems Solved by ML: In *Data Processing4VIS*, ML techniques demonstrate the ability to make raw data better satisfy certain visualization-related purposes, including enabling efficient visualization creation and enhancing human perception.
▸ *Enabling efficient visualization creation.* By narrowing down the purpose of data processing to the creation of specific visualizations, the employment of ML techniques can improve the efficiency of

TABLE 3
Seven main visualization processes that benefit from employing ML techniques.

	Process	Description	Problems Solved by ML	Representative Examples
DATA	Data Processing4VIS	raw data is transformed into a format that better suits the following visualization processes	improve the efficiency of data processing	ML cleans data for a specified visualization by asking questions to users [82]
			enhance perception of generated visualizations	ML transforms high dimension data to 2D and maximizes the human-perceived separation among classes [29]
VIS	Data-VIS Mapping	data fields are mapped into visual channels	generate suitable visualizations	ML generates Vega-Lite visualization specifications for a given dataset [41]
			improve the efficiency of visualization creation	ML synthesizes intermediate density map images between given density maps without storing and visualizing data of all time steps [48]
	Insight Communication	insights are embedded in visualizations to be effectively communicated	interpret insights	ML recognizes entities from the user-provided text insights [59]
			generate suitable visualizations	ML predicts suitable visualization specifications for communicating given insights [31]
	Style Imitation	styles are extracted from the given examples and applied to the created visualization	imitate the color selection	ML generates color ramps by imitating designers' practices in choosing colors [62]
			imitate the layouts	ML generates graph drawings with similar layout styles of the given examples [50]
USER	VIS Interaction	users interact with a visualization and transformed it into a new stage through user actions	refine the result of the current interaction	ML refines the selected points of a lasso selection in 3D point clouds [46]
			understand natural interaction	ML interprets the given text descriptions and extracts the corresponding visual patterns [25]
	User Profiling	user actions with visualizations are logged and analyzed to better understand users	predict user behavior	ML predicts a user's next click in a scatter plot [66]
			predict user characteristics	ML predicts a user's task performance and personality based on mouse interaction [6]
	VIS Reading	users read visualizations and obtain useful information	extract content	ML extracts an extensible visualization template from a timeline infographics [60]
			interpret content	ML answers questions about a given bar chart visualization [38]
			estimate human perception	ML predicts the distribution of users' attention on an infographic [18]

data processing. For example, data cleaning, an inevitable step in data processing, is often treated as an independent step and completed before the visualization creation. Instead, Luo et al. [82] related the data cleaning processes with visualization creations and used ML techniques to help users interactively clean the data after the visualization creation. Fig. 2(b) illustrates an overview of [82]. Users first specify a visualization and create it using the uncleaned data. A trained model then detects errors and generates cleaning options for the data underlying the visualization. These cleaning options are provided to users in the form of yes/no questions. Based on users' answers, the underlying data is cleaned and the visualization is updated. This progressive data cleaning enables a more flexible and efficient creation of visualization, since data does not need to be totally cleaned before creating visualizations.
▸ *Enhancing human perception.* Meanwhile, ML techniques also demonstrate the ability to enhance user perception of the created visualizations in *Data Processing4VIS*. For example, dimension reduction (DR) is a widely used data processing method for visualizing high dimensional data. Previous studies often directly employ the general DR methods proposed in ML field without considering the needs in visualizations. To better present data in scatter plots, Wang et al. [29] proposed a supervised DR method that takes the perceptual capabilities of humans into account. The modified DR method mimics human perception in cluster

separation and successfully maximizes the visually perceived cluster separation in 2D projections.

4.2 Data-VIS Mapping

Process Description: In *Data-VIS Mapping* (Fig. 3(a)), the values of data fields are mapped into the visual channels of graphic marks. Appropriate visual mapping is needed in this process to help people better understand and analyze the visualized data. Such a mapping is usually manually specified using code or authoring tools, which results in steep learning curves and makes data visualization inaccessible to general users.

Problems Solved by ML: In the collected ML4VIS studies, ML techniques mainly facilitate *Data-VIS Mapping* by recommending suitable visual representations and by improving the efficiency of visualization creation.
▸ *Recommending visualizations.* A set of recent research has shown that ML can be used to automatically recommend suitable visual representations. For example, DeepEye [35] combines supervised ML techniques with expert rules to automatically recommend good visualizations for users. Given a dataset, all the possible visualizations are enumerated and classified as "good" or "bad" by a binary decision tree classifier. All the "good" visualizations are then ranked and provided to users. Meanwhile, the great success
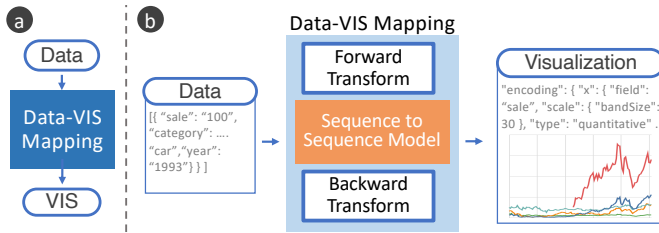
Fig. 3. (a) An illustration of the *Data-VIS Mapping* process. (b) A representative example of employing ML in *Data-VIS Mapping*: Data2Vis [41]. An ML model (orange box) is trained to map JSON-encoded datasets into Vega-lite visualization specifications.
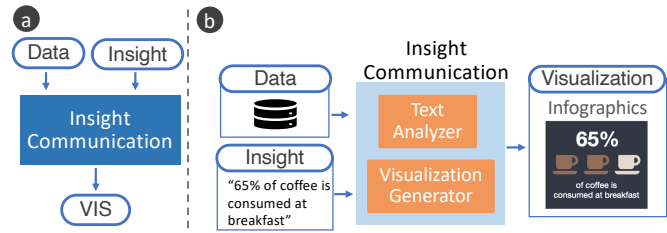

Fig. 4. (a) An illustration of the process of *Insight Communication*. (b) One representative example of ML4VIS in *Insight Communication*: Text-to-Viz [59]. A text insight is first analyzed by a text analyzer and then be transformed into expressive infographics by a visualization generator.

of deep learning in various domains promotes its application in *Data-VIS Mapping*. VizML [69] collects a large visualization dataset and trains a fully connected neural network to predict the top five design choices when creating visualizations for a specific dataset. Data2Vis [41] formalizes the process of *Data-VIS Mapping* as a sequence to sequence translation from the original data to the visualization specifications. As illustrated in Fig. 3(b), a sequence-to-sequence neural network learns the mapping between JSON-encoded datasets (source sequence) and Vega-lite visualization specifications (target sequence). In these studies, ML automatically learns design choices from the training dataset and directly translates data to suitable visualizations. As a result, the employment of ML can effectively reduce the manual efforts in selecting data attributes, designing visual representations, and specifying the mapping from data values to graphic marks.

▸ *Improving the efficiency of visualization creation.* Another line of research focuses on improving the efficiency of creating visualizations. One direction is to improve the efficiency of creating dynamic visualizations. For instance, creating dynamic density maps usually leads to high computational cost and memory demand since a large amount of data needs to be recorded and then visualized at each time step. To address this issue, GenerativeMap [48] proposes a generative model to synthesizes a series of density map images and show the dynamic evolution between two given density maps, thus relieving the burden of storing intermediate results. Similarly, to visualize the dynamics in spatial-temporal data, TSR-TVD [47] applies a recurrent generative network to learn the pattern in dynamic evolution and generate temporal super-resolution visualizations. In these studies, ML learns the patterns in dynamic evolution from training data and directly synthesize intermediate visualizations to reduce the needed storage and computation cost. Another interesting direction is to facilitate parameter exploration in visualization creation. For example, Berger et al. [28] trained a Generative Adversarial Network (GAN) to synthesize visualizations and guide users in transfer function editing by quantifying expected changes in the visualization. Similarly, InsituNet [45] introduces a deep learning based surrogate model to create visualizations of simulation data by learning the mapping from simulation parameters to visualization images. These methods enable people to explore synthesized visualizations under various input parameters without actually executing the expensive creation processes.

4.3 Insight Communication

Process Description: In *Insight Communication* (Fig. 4(a)), insights are transformed into visualizations that can effectively communicate them. As with many studies in the visualization

community [117], [118], we refer to the term *Insight* as the knowledge about the data that is communicated through visualizations. The main difference between *Insight Communication* and *Data-VIS Mapping* is the availability of insights. In *Data-VIS Mapping*, insights are hidden in the underlying data and need to be discovered by people through visual exploration. In *Insight Communication*, on the contrary, insights are already available to designers and need to be highlighted in the created visualizations. This difference is also denoted as exploratory versus explanatory visualization in prior studies [119]. To better communicate insights, visualizations in *Insight Communication* are usually more visually pleasing to improve user engagement and memorability.

Problems Solved by ML: In *Insight Communication*, existing studies have successfully applied ML techniques to interpret insights and to build a mapping from insights to suitable visualizations.

▸ *Interpreting insights.* The insights can be either provided by the designers or automatically generated from the data. For example, Text-to-Viz [59] allows users to provide insights through natural language statements such as *"40% of USA freshwater is for agriculture"* (Fig. 4(b)). A text analyzer (a supervised CNN+CRF model) is trained to extract entities from the sentence and understand the provided insights. DataShot [58] automatically identifies insights from tabular data and presents these identified insights to the users through infographics.

▸ *Mapping insights into visualizations.* ML techniques also demonstrate the capability to learn the mapping from insights to suitable visual representations. To facilitate this mapping, ML4VIS studies usually describe visual representations using specifications in a predefined design space to ensure a well-defined output space. Using the information from insights, ML can identify candidates for each dimension in the design space and produce the final visualization from these valid combinations [31], [58], [59]. For example, DeepEye [31] employs declarative visualization languages similar to Vega-Lite to enable the creation of common visualizations (*e.g.*, bar, pie charts). A decision tree model learns the mapping from insights and data characteristics to visualization specifications. Meanwhile, to generate visualizations with higher aesthetic values in *Insight Communication*, researchers also contribute more advanced design spaces. For example, Text-to-Viz [59] summarizes four design-space dimensions of infographics–layout, description, graphic, and color–from discussion with design experts. A visualization generator synthesizes infographics by generating a tuple of values on these four dimensions.

4.4 Style Imitation

Process Description: In *Style Imitation* (Fig. 5(a)), the styles of given visualizations examples are applied to create new visualiza-
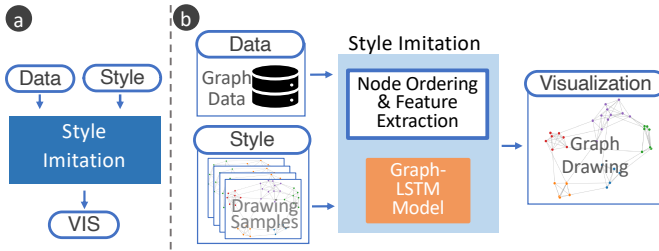
Fig. 5. (a) An illustration of the process of *Style Imitation*. (b) A representative example of ML4VIS in *Style Imitation*: DeepDrawing [50]. A Graph-LSTM model (orange box) learns the drawing styles from a set of samples and generates drawings of a similar style for the input data.

Fig. 6. (a) An illustration of the *VIS Interaction* Process. (b) A representative example of employing ML in *VIS Interaction*: LassoNet [46]. An ML model (orange box) predicts the user intended node selection in a 3D point cloud from a lasso drawn on 2D space.

tions. We use the term *style* to denote a set of variables that affect the appearance of visualizations but are not directly mapped to data values. Typical visualization styles include the color palette, the chart decoration, and the view layout and aspect ratio. The main difference between *Data2VIS Mapping* and *Style Imitation* is whether the visual properties are mapped to data values. For example, *Data-VIS Mapping* decides whether data should be visualized as a scatter plot or a bar chart. In contrast, *Style Imitation* extracts color palettes from given examples for creating a scatter plot.

Problems Solved by ML: Even though the visualization style involves a large number of variables, existing ML4VIS studies on *Style Imitation* mainly apply ML techniques on two aspects, the imitation of color selection and the imitation of layout styles.

▸ *Imitating color selection.* Color selection in visualizations is not only an aesthetic choice but also influences the effectiveness of information communication. Even though some helpful guidelines about colors are available, it is still challenging for novice visualization developers to use these loosely defined qualitative guidelines to generate high-quality color ramps. To address this issue, Color Crafter [62] mimics the practices of professional designers and automatically generates high-quality color ramps. The color style imitation is achieved by modeling the paths that expert-designed color ramps traverse through color space. The authors trained ML models in a corpus of 222 expert-designed color ramps that are summarized from popular visualizations, and used these models to generate effective color ramps for developers from one single seed color.

▸ *Imitating layouts.* A parallel line of research on *Style Imitation* focuses on generating visualizations of similar layouts.

ML techniques have been extensively employed to generate graph drawings that mimic the layout styles of given graph drawing examples. Kwon et al. [17] presented an ML approach to facilitate large graph visualization by learning the topological similarity between large graphs. Specifically, the method provides a quick overview of visualizations for an input large graph data by learning from the drawings with similar topology structures. This approach is further extended in a recent study [49], which learns the layout from examples using a deep generative model. The model is then used to provide users with an intuitive way to explore the layout design space of the input data. Meanwhile, DeepDrawing [50] trains a graph-LSTM to learn one specific layout style from graph drawing examples. Instead of drawing a single graph in diverse layouts, the trained ML model in DeepDrawing directly maps new input data into graph drawings that share similar layout styles with the training examples, as shown in Fig. 5(b).

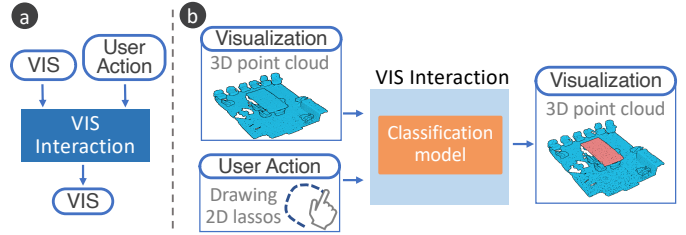Apart from graph drawing, a few studies have explored imitating the layouts of other types of visualizations, including storyline visualization [75] and mobile visualization [74]. For example, PlotThread [75] trains a reinforcement learning agent to learn the layout of human-designed storyline visualizations. By imitating the layout from human-drawn examples, PlotThread is able to include more diverse narrative elements and create more expressive storylines compared with optimization-based methods.

4.5 VIS Interaction

Process Description: In *VIS Interaction* (Fig. 6(a)), the appearance of a visualization is modified based on the user actions (*e.g.*, zoom, filter). *VIS Interaction* usually reflects user intentions in visual exploration and modifies the visualization by manipulating the view configurations, the visual mappings, or the underlying data [120].

Problems Solved by ML: In *VIS Interaction*, previous studies have demonstrated the capabilities of ML techniques to infer user intentions and thus assist users in interactions. According to our survey, the visualization problems solve by ML techniques in *VIS Interaction* can be classified into two main groups, refining interaction results and understanding natural interactions.

▸ *Refining interaction results.* ML techniques can refine the results of user interactions to achieve more accurate and efficient interactions. When interacting with a large number of visual elements, an accurate and efficient selection is crucial but usually hard to achieve. Take the brushing in 2D space as an example. Traditional interactions often need to make a trade-off between interaction efficiency and accuracy, *e.g.*, rectangular brushing is fast but inaccurate while the logical combination is accurate but slow. Fan and Hauser [36] exploited a CNN model to achieve both fast and accurate node selection in 2D scatter plots. The CNN estimates the intended node selection based on a simple line-brushing and data distribution in the visualization. Compared with selections in 2D space, node selection in 3D space is even more challenging: visualizing 3D points in 2D space easily causes occlusion; input devices such as touchscreens only operate in 2D space. To facilitate the node selection in the large-scale 3D point cloud, LassoNet [46] uses a deep learning model to predict the node selection based on a lasso drawn on a 2D surface, the user's viewpoint, and characteristics of the point cloud. A pipeline of LassoNet is illustrated in Fig. 6(b).

▸ *Understanding natural interactions.* Natural interactions are interactions that humans naturally communicate through, such as gestures, natural language, and sketches [121]. ML techniques have demonstrated the capabilities to understand nature interactions and allow a more intuitive and convenient way to interact with visualization. Given that the input natural interaction (*e.g.*, sentence,

sketch) can be fuzzy and uncertain, ML techniques help interpret the user's input and trigger dedicated functions. For example, ShapeSearch [25] allows users to search visual patterns from trend line visualizations using sketches and text descriptions. The authors trained a CRF model to recognize entities and understand natural interactions. FlowSense [44] supports users in visual data exploration via natural language command, *e.g.*, *"highlight the selected cars in a parallel coordinates plot"*. While the language understanding is mainly accomplished by a non-ML method (semantic parsing) in FlowSense, a supervised learning model is trained to effectively resolve syntactic ambiguity in the language commands.

4.6 User Profiling

Process Description: In *User Profiling* (Fig. 7(a)), user actions with visualizations are logged and then analyzed in order to better understand users.

Problems Solved by ML: In *User Profiling*, previous studies have demonstrated the capabilities of ML models in modeling user behaviors and characteristics.

▶ *Predicting user behaviors.* ML techniques can predict next-step interactions and facilitate complex visual explorations that consist of a series of interactions. Therefore, the employment of ML can ease the learning curve for users or reduce the interaction latency in complex visual explorations. For example, to ease the learning curve, REACT [32] models the analysis context using a generic tree-based model, where the edges represent the user's actions and the nodes represent the system states. Given an analysis context, relevant contexts are first fetched using a context-similarity metric and then used to generate next-action suggestions for the user. Another example is the study by Ottley et al. [66]. To reduce the interaction latency, Ottley et al. utilized a Markov model to predict the user's next mouse clicks in large scatter plots and fetch related data in advance.

▶ *Predicting user characteristics.* Interactions with the visualization can reflect users' reasoning processes in visual exploration and even their own personal characteristics. ML can effectively analyze these interactions and help designers learn about the users, including their learning abilities [7], their cognitive abilities [5], their analysis goals [2], [26], and even their personalities [6]. For example, Gramaz et al. [26] demonstrated that the classification of mouse interactions can effectively infer the visual analysis tasks of cancer genomics experts. The ML classification can even expand current knowledge about the visualization tasks in cancer genomics. Brown et al. [6] applied ML techniques to analyze users' mouse interactions in the visual exploration of finding Waldo. They found that ML can accurately predict the user task performance (*i.e.*, the time used to find Waldo) and uncover user personalities (*e.g.*, locus of control, extraversion, and neuroticism). Apart from mouse interactions, eye gaze data has also been treated as a type of interaction data and can help learn about users [122]. Lallé et al [7], used ML techniques to predict a user's learning curve of a visualization based on the eye gaze data. By learning about users, visualization developers are able to go beyond the one-size-fits-all design and provide adaptive visualizations for different task needs, user abilities, and analysis stages.

4.7 VIS Reading

Process Description: In *VIS Reading* (Fig. 8(a)), users observe the appearance of a visualization, read the encoded data, and
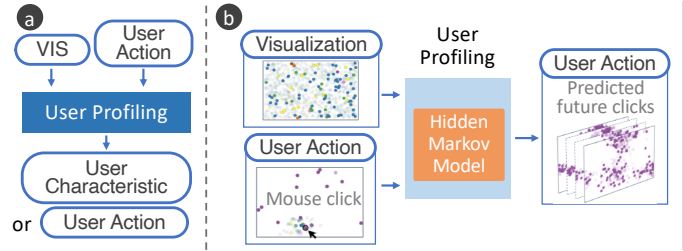

Fig. 7. (a) An illustration of the *User Profiling* process. (b) A representative example of employing ML in *User Profiling*: Ottley et al. [66] employed a hidden Markov model (orange box) to predict user future actions based on their past actions and the visualization.
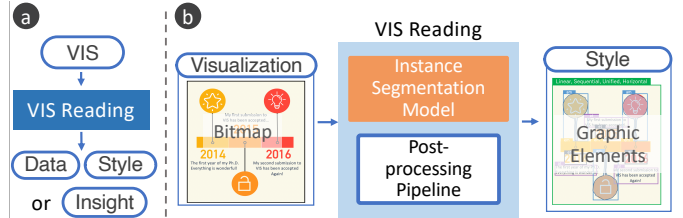

Fig. 8. (a) An illustration of the *VIS Reading* process. (b) A representative example of employing ML in *VIS Reading*. Chen et al. [60] employed a deep learning model (orange box) to extract the graphical elements from a timeline infographics.

understand the underlying information. ML techniques have been used in *VIS Reading* process to automatically *"read"* the visualizations like humans, thus helping designers better understand users and enabling the analysis of a large corpus of visualizations.

Problems Solved by ML: In *VIS Reading*, the capabilities of ML techniques can be divided into three main groups: extracting content, understanding content, and modeling users.

▶ *Extracting content.* ML techniques can directly extract content from visualizations in the form of bitmap images, such as graphical elements [60], color palette [16], and visual encodings [21]. The extracted content can be used for indexing, restyling, and reusing visualizations. A typical example is [21], which contributes an end-to-end pipeline to extract Vega visualization specifications from standard visualizations such as line charts. A set of classification models are trained to identify chart types, localize text, and recognize graphic marks separately. Compared with standard visualizations, infographics have more diverse appearances and are more difficult to interpret. Chen et al. [60] utilized a deep learning model, Mask RCNN, for instance, segmentation to extract graphical elements from infographics. The extracted graphical elements are used as visualization templates for creating similar infographics using different data.

Meanwhile, since contents of bitmap visualizations are usually hard to be automatically extracted, some studies have proposed methods to embed the required information (*e.g.*, meta data, color schema) into bitmap visualizations [73], [123]. The embedded information will not influence human perception of the visualization and can then be easily extracted from the visualization. For example, VisCode [73] trains an encoder-decoder network, in which the encoder embeds a QR code to the background of a bitmap visualization and the decoder extracts the QR code from the visualization. A BASNet is trained to identify the semantically important regions and ensures the encoder will not affect human

Fig. 9. (a) A visualization pipeline proposed by Card [104]. (b) A model of visualization proposed by Van Wijk [103].

Fig. 10. An ML4VIS pipeline connects the seven visualization processes that can be assisted by ML. Each blue rectangle presents a visualization process where ML can assist in. Each white circle indicates a key element and can be either the input or the output of a process. The arrows connecting the rectangles and circles indicate the input and output of each process.

Fig. 11. The employment of ML in one visualization process can benefit other processes and further improve the whole workflow. Analyzed examples: A) Bylinskii et al. [18] employed ML in *VIS Reading* and further improved a manual *Insight Communication* process. B) Ottley et al. [66] employed ML in *User Profiling* and further improved an non-ML assisted *VIS Interaction* process.

5 AN ML4VIS PIPELINE

To map out the role of ML4VIS in general visualization and better inform future exploration, we connect the seven processes to previous visualization models and present an ML4VIS pipeline.

This ML4VIS pipeline is inspired by existing visualization models, especially the one proposed by Card [104] and the one by Van Wijk [103] (Fig. 9). We modify these visualization models to better fit the scenarios of ML4VIS, *i.e.*, to describe the visualization processes so that they can be better aligned with ML tasks. Specifically, we make the following main modifications. First, contrary to previous studies, we take an additional element "style" into consideration. We refer to *style* as elements that affect the appearance of a visualization but are not mapped to data values. Typical visualization styles include color palette, layout, and chart decoration. *Style* is tightly related to the memorability [124] and engagement [125] of a visualization and has gained increasing research attention. Second, we decouple this pipeline from users' internal states, *e.g.*, users' internal knowledge in Van Wijk's model, to better fit the context of ML4VIS. Instead of including users' internal states, we use the outputs of these internal states (*e.g.*, insights, user action) to ensure elements in the pipeline are describable instances rather than vague concepts. For example, user actions can reflect users' intentions and tasks and insights can reflect users' perceived knowledge from a visualization. Since ML usually requires well-defined inputs and outputs, the modified pipeline can better fit the context of ML4VIS by removing vague or hard-to-define concepts. Meanwhile, as in Van Wijk's pipeline, we also do not differentiate *visual mapping* and *view transformation* that are introduced in Card's pipeline. This decision is based on our observation of the collection of ML4VIS studies. In the collected ML4VIS studies, the visual mapping and the view transformation are sometimes achieved by an end-to-end ML model and are not independent (*e.g.*, [47], [48]).

The ML4VIS pipeline is presented in Fig. 10. Each of the seven blue boxes indicates a process where ML can assist in; each of the white boxes indicates a key element that is either the input or the output of a process. As with Van Wijk's model [103], the seven processes are grouped by three containers: *DATA*, *VIS*, and *USER*. **Example Usage:** This ML4VIS pipeline illustrates the close relationship among the seven processes: the outputs of one process are used as the inputs for other processes. Therefore, employing ML techniques in one visualization process can further facilitate other processes, even though ML techniques are not used in these processes. When designing visualization tools, this ML4VIS pipeline can help decide where to embed ML techniques to enhance the whole workflow. Below we present several examples in the collected papers as a demonstration.

perception of these regions.

▸ *Interpreting content.* More than extracting content, ML techniques can also achieve a high-level interpretation of visualizations and solve complex tasks. For example, Kembhavi et al. [13] proposed an LSTM-based method to understand the concepts and relationships in a diagram. Questions about the diagram like *"will the increases in lion lead to the decreases in deer?"* can be answered by an attention-based model with high accuracy. Another recent example DVQA is presented by Kafle et al. [38]. DVQA combines a CNN-based and an LSTM-based network and can answer questions about data visualizations. Given a visualization, the model reads the visualization content and answers reasoning questions such as *"which item sold the most units?"*

▸ *Estimating human perception.* ML techniques can model how humans perceive a visualization and help designers better assess visualizations and understand audiences. A number of ML-based methods have been contributed to model user perception of a visualization from various aspects, including visual attention [18], similarity recognition [57], and readability assessment [42]. For example, Bylinskii et al. [18] trained a deep learning model to predict the distribution of users' visual attention on an infographic. Guided by this prediction, designers can better arrange the layout and ensure that the most important contents are emphasized in the infographic. Haleem et al. [42] proposed a CNN-based approach to evaluate the readability of graph layouts by using graph images as inputs. An interesting phenomenon is that current ML4VIS studies extensively study the perception modeling in scatter plots, such as the perception of class separation [11], the perception of local patterns [37], and the perception of similarity [57]. This focus on scatter plots might stem from the wide popularity of scatter plots as well as their relatively simple visual forms.

- Bylinskii et al. [18] proposed a deep learning method to estimate the visual attention of audiences on different regions of an infographic. The employed ML technique takes visualization as input, estimates user attention, and predicts user-perceived insights (*i.e.*, information in the highlighted regions), thus contributing to *User Profiling*. Based on this method, the authors developed a tool to help designers evaluate their visualizations by checking whether important information received sufficient attention. Meanwhile, the outputted insights can be used as input of *Insight Communication*. In the developed authoring tool, designers can also interactively modify their infographic, observe the change of audience attention, and thus create visualizations that better communicate important insights. As shown in Fig. 11(A), the developed tool forms an *Insight Communication-VIS Reading* loop: the *Insight Communication* process does not employ ML techniques but benefits from the ML techniques used in *VIS Reading*.

- Ottley et al [66] applied a Markov model to learn users' needs based on their mouse clicks and predicted users' next-step selections. While the ML techniques were employed in *User Profiling*, the predicted next-step selections can be used as input to the *VIS Interaction* and help the visualization tool fetch potentially related data in advance, thereby improving the interaction latency in large-scale data analysis. As a result, the *VIS Interaction* process, even though employing no ML techniques, can benefit from the ML techniques used in *User Profiling*. As shown in Fig. 11(B), *VIS Interaction* and *User Profiling* form a loop.

Meanwhile, the unconnected items in this pipeline can illustrate possible future research directions. For example, the user characteristics extracted from *User Profiling* can be used as inputs of *Data-VIS Mapping*, *Insight Communication*, and *Style Imitation* to construct user-adaptive visualizations, *i.e.*, visualizations that consider an individual user's needs, abilities, and preferences. While several studies [5], [7], [8] have discussed this direction in their future work, few ML techniques have been employed to successfully achieve this goal [2].

6   ALIGN ML WITH VIS

In this section, we review the current ML4VIS studies from an ML perspective, aiming to answer the question *"how is ML used to solve visualization problems"*. We analyze both the training data and the ML tasks. We first discuss the commonly-used formats of training data, which are summarized from the collected 88 ML4VIS studies using a bottom-up approach. We then categorize the collected ML4VIS papers based on the main ML tasks [115], [116] to provide an overview of how the needs in visualization are formed and solved as ML problems. This ML perspective (*i.e.*, training data, ML tasks) naturally corresponds to the visualization perspective (*i.e.*, process inputs and outputs, visualization processes). We map the summarized data formats to the six types of information (*i.e.*, inputs and outputs of the visualization processes) and the ML tasks to the seven visualization processes. This ML-VIS mapping aims to provide both an understanding of the current practices and a guidance for future exploration in the ML4VIS research.

6.1   Training Data Formats

The format of training data reflects how information is encoded in ML. It can influence how to formalize ML problems and choose suitable ML models. The seven visualization processes in Section 4 reveal six types of information: data (underlying the visualization), visualization, user action, user characteristics, style, and insight. In this subsection, we summarize the common data formats that are used to encode the five types of information based on a bottom-up analysis of the collected papers (as shown in Table 4).

Image is the format that most visualizations "in the wild" are presented in [3]. It is not surprising that many studies (*e.g.*, [16], [21], [22], [23]) encode **visualizations** in the form of images, contributing a number of large datasets of real-world visualization images: FigureSeer Dataset (60k) [12], AI2D dataset (5k) [13], Visually29K dataset (29k) [43], DVQA dataset (300k) [38], FigureQA dataset (100k) [34], ColorMapping dataset (1.6k) [16]. The emergence of large visualization-image datasets may result from the considerable differences between natural images and visualization images. Such differences make it difficult to directly apply the datasets and techniques developed for natural images to visualization images [13], [39], [60]. The **styles** of visualizations and the **insights** extracted from visualizations can also be encoded as parts of an image. For example, Chen et al. [60] encoded styles of a visualization as its segmented graphical icons. Bylinskii et al. [23] encoded the insights of an infographic as visual elements that are diagnostic of the topic of an infographic. Meanwhile, some researchers encode **user actions** in images. For example, Fan et al. [36] encoded the user actions (*i.e.*, click and drag) in a scatter plot as the zoom and rotation of the scatter plot image.

Natural language (*e.g.*, text and speech) is the most common form of human communication [126]. Prior studies [44], [63] have enabled users to use natural language to express intended **user actions**, aiming to achieve more convenient and intuitive *VIS Interaction*. For example, Huang et al. [63] allowed users to interact with a visualization system using textual sentences such as "visualize trajectories passed through tourist attractions

TABLE 4
Examples of Different Training Data Formats.

	Data	Visualization	User Action	User Characteristic	Style	Insight
Image	—	[12], [16], [21], [22], [23], [34], [38], [45], [48], [60], [73]	[36], [70]	—	[60], [87]	[23], [43], [79]
Natural Language	—	—	[44], [63]	—	—	[22], [38], [39], [59], [81]
Engineered Features	[35], [69]	[3], [40]	[5], [7], [26]	[7], [8]	[42]	—
Sequences	[41]	[41], [88]	[6], [32], [52], [66]	—	—	—
Graph	[49], [50]	—	—	—	—	[13], [39]

during January 25". Apart from user actions, **insights** can also be encoded in the form of natural language. Through natural language, users can easily describe the insights they aim to convey in *Insight Communication* [59] and understand the insights extracted in *VIS Reading* [13], [38], [81]. For example, Text-to-Viz [59] allows users to describe the insights using a natural language statement and automatically generates the corresponding infographics. AutoCaption [81] generates a text caption to describe the insights of a visualization based on four types of chart features detected (*i.e.*, aggregation, comparison, trend, distribution). In contrast to images, researchers usually collected a small number of labeled natural language examples for their specific tasks. For example, Yu and Silva [44] collected and labeled less than 20 examples to train an ML model that can resolve certain syntactic ambiguity; Cui et al. [59] collected and labeled 800 examples to train an ML model that can extract the four predefined entities. Also, due to the small number of training data, ML-based methods are often combined with non-ML-based methods. For example, Yu and Silva [44] supported natural language queries using non-ML-based semantic parsing. An ML model is only trained on few labeled examples to resolve certain syntactic ambiguities.

Engineered features are high dimensional vectors that are extracted from the raw data using domain knowledge. While the great progress of deep learning helps reduce the reliance on human-centric feature engineering, extracting features using human knowledge is still popular in ML4VIS studies. This popularity of feature engineering may come from its ability to effectively capture important information without the need for large datasets and enormous computational resources. Human specified features have been used to encode all the five types of information except insights. Here we discuss several representative examples. For the **data** (underlying a visualization), VizML [69] encodes a tabular data using 841 features related to statistical analysis, such as mean, standard deviation, entropy, and skewness. For the **visualization**, Beagle [40] encodes visualizations using the features extracted from SVG visualization files, such as the number of axes, the position of circles, and the width of rectangles. For the **user actions**, Gramazio et al. [26] encoded user actions (*i.e.*, mouse clicks) through a set of human-defined features, such as the dwell time, the active time, the region-of-interest transitions. For the **style**, Haleem et al. [42] summarized the style of a graph layout using a set of graph aesthetic metrics, such as the node occlusion and the edge crossing.

Sequences are rows of values whose order is important. Sequences have been widely used to encode **user actions** in their temporal order [6], [52], [66]. For example, Brown et al. [6] conducted a sequence-based analysis of the user actions in visual analytics. The sequences of seven types of user actions (*i.e.*, pan left, right, up, down, zoom in, and out) are converted into sub-sequences of two or three user actions, such as first zoom in and then zoom out. Interestingly, even though **data** (underlying visualizations) and **visualizations** do not have a sequential structure, they can also be encoded using sequences. For example, Data2Vis [41] encodes both JSON data file and Vega-lite visualization configuration as sequences. Table2Chart [88] encodes a visualization as a sequence of actions, including selecting data fields, selecting chart types, splitting data fields, and grouping data fields.

Graph is a non-Euclidean data format that consists of nodes and edges. **Data** (underlying visualizations) can be encoded in its original graph format in ML4VIS studies [49], [50]. Meanwhile, **insights** of visualizations can also be encoded in the form of graphs [13], [39]. For example, Kim et al. [39] encoded the insight
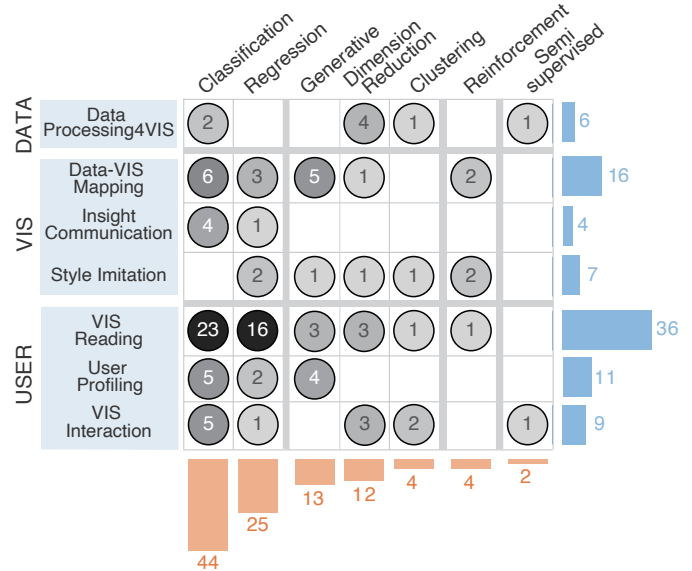
Fig. 12. Each row is a visualization process and each column is an ML task. In each cell, the number and the grayscale indicate the number of papers. The height of the bar indicates the number of studies related to a type of ML task (blue bar) or a visualization process (orange bar). Note that the number on a bar can be smaller than the sum of the numbers in the corresponding row/column, because some papers involve multiple learning tasks or visualization processes.

of an infographic as a graph. Each node of the graph represents an object in the infographic and each edge represents the relationship between the two connected objects.

6.2 Mapping ML Tasks to Visualization Processes

This subsection first provides a statistic summary of how the different ML tasks are mapped into different visualization processes. Then, we discuss each learning task in detail, including the related visualization processes, the commonly-used ML models, and the representative examples.

When summarizing the collected studies from an ML perspective, we find that it is not practical to summarize based on the employed ML models, as new ML models are constantly emerging and one ML model is often applied to solve different problems. To better align the capabilities of ML with the needs in visualization, we categorize the existing ML4VIS studies based on the types of ML tasks. This learning-task-based categorization allows us to understand how the needs in visualizations are formed and solved as ML problems, without diving into the ML model details (*e.g.*, whether to use LSTM or GRU).

Current ML methods can be categorized into four basic paradigms, *i.e.*, supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning [115], [116]. Based on the collected 88 papers, we refer to the learning tasks discussed in [115], [116] and further categorize the four paradigms into seven subsets: clustering, dimension reduction, generative, classification, regression, semi-supervised learning, and reinforcement learning. One subset of learning tasks is obtained when the following categorization is no longer related to learning tasks. For example, Nicolas [115] further categorizes reinforcement learning as Markovian or evolutionary. Since this categorization is about learning mechanisms rather than learning tasks, we treat reinforcement learning as one minimal subset of learning tasks.

6.2.1 An Overview

Fig. 12 shows the ML-VIS mapping between the seven visualization processes and the ML tasks. Each column represents one type of learning task and each row represents one visualization process. The columns are grouped using gray thick lines based on the four paradigms, *i.e.*, supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. The four-column groups, and columns inside each group, are in descending order of the number of papers. In each cell, the number and the grayscale indicate the number of studies that are related to both the ML task in column and the visualizations process in row. The height of the bar indicates the number of studies related to a type of ML task (orange bar) or a visualization process (blue bar). Note that the number on a bar can be smaller than the sum of the numbers in corresponding cells, since a paper can involve more than one visualization process and one ML task.

Overall, most ML4VIS studies formulate the needs in visualization processes as supervised learning tasks (44 as classification and 25 as regression). Other types of learning tasks are less used. Unsupervised learning, the second most-used learning, is employed in 27/88 ML4VIS studies. Only 4 papers use reinforcement learning and 2 papers use semi-supervised learning. This is not surprising as supervised learning is the most common type of learning [116]. When taking different visualization processes into consideration, the preference towards supervised learning, especially classification, appears on most visualization processes. This preference is most pronounced in *VIS Reading*, where 23/36 studies are related to classification. Unlike other visualization processes, in *Data Processing4VIS* and *Style Imitation*, the collected ML4VIS studies exhibit a preference for unsupervised learning. But this observation needs to be treated cautiously due to the small number of studies in the two processes.

Among the seven visualization processes, *VIS Reading* is the most frequently investigated visualization process in ML4VIS (by 36/88 papers). Most of the studies in *VIS Reading* employed supervised learning (23/36 used classification and 16/36 used regression). We speculate this is caused by the success of deep learning in computer vision tasks (*e.g.*, image classification, object detection, instance segmentation). A large number of deep learning models have been proposed to understand natural images and are adopted for the perception of visualization images [18], [43], [60], [85], [87]. A considerable proportion of ML4VIS studies investigate the problems in *Data-VIS Mapping* (16 papers), *User Profiling* (11 papers), and *VIS Interaction* (9 papers). The other three processes (*i.e.*, *Data Processing4VIS*, *Insight Communication*, *Style Imitation*) are relatively less investigated.

6.2.2 Supervised Learning

In supervised learning, a model learns the mapping from input X to output Y from the labeled training examples. Labeled input-output pairs are required in supervised learning. Supervised learning can be further divided into **regression** and **classification** based on whether the output is numerical or categorical [116].

When formalizing a visualization problem as a **regression** task, the training examples need to be labeled with numerical values. Among all visualization processes, regression is mainly (19/25) used in the USER stage, especially in *VIS Reading* (16 papers). For example, in *VIS Reading*, regression can be used to learn the salient regions of a visualization [18], [43], the segmentation of important visual objects [60], and the aesthetic scores of visualizations [17], [42]. In *User Profiling*, regression can also be used to learn the

dwell time of user mouse clicks [26] or the users' learning curve of understanding a visualization [7]. Regression is not used in DATA stage and less used in the VIS stage, which may be due to the difficulty to quantitatively label the data processing and the visualization creation.

Many ML models can be used for regression tasks. How to choose a suitable ML model largely depends on the data formats. For example, CNNs are widely used when the input data are images. Many CNN models that are originally proposed for general computer vision tasks have been adopted to process visualization images, including ResNet (used in [12]), Xception (used in [79]), FCN used in ([127]), Fast R-CNN (used in [43]), and Mask R-CNN (used in [60], [85]). LSTM and GRU are used when processing sequences and natural languages [38], [39]. Traditional (*i.e.*, non-deep-learning-based) ML models (*e.g.*, SVR, logistic regression, MLP) are usually used when the inputs are engineered features.

When formalizing a visualization problem as a **classification** task, all the training examples need to be categorized into finite classes. Classification has been widely used in ML4VIS studies, such as evaluating whether a visual encoding is valid [35], predicting whether a visual region has interesting patterns [83], and recognizing the types of charts [3], [22]. Complicated problems can be decomposed into a series of classification tasks. For example, VizML [69] decomposes the creation of a visualization as a series of classification tasks, including classifying the type of marks, the type of shared axis, and the type of visualization. Kim et al. [39] and Kembhavi et al. [13] decompose the understanding of a diagram infographic as classifying the relationships between every two visual objects in the diagram. Classification has been used in all the seven visualization processes except for *Style Imitation*. We conjecture this is because that the possible styles of a visualization can hardly be categorized into finite classes.

Most models used for regression can also be used for classification tasks through different ways, for example, setting a threshold (binary classification, 1-vs-the-rest classification) or using a softmax function (multi-class classification). Meanwhile, there are some ML models whose outputs are, by nature, discrete, such as SVM and decision tree.

The recent success of end-to-end deep learning methods has contributed many ML models that can accomplish regression and classification tasks at the same time. For example, Faster RCNN [128], an instance segmentation model, has been used in VIS Reading and can predict the type (classification) and the bounding box (regression) of visual objects simultaneously [43].

6.2.3 Unsupervised Learning

In unsupervised learning, a model learns the underlying structure of the unlabeled data X. Compared with supervised learning, unsupervised learning relieves the requirement of the expensive data labeling process. Based on the collected papers, we discuss the three common tasks when using unsupervised learning in ML4VIS: generative learning, dimension reduction, and clustering.

Similar to prior research [49], this paper refers a **generative** model to a model that learns the distribution of unlabeled data and is capable of generating new samples that are similar to, but not the same as, the training data. Generative learning is mainly used in four visualization processes: *Data-VIS Mapping*, *Style Imitation*, *VIS Reading*, and *User Profiling*. In *Data-VIS Mapping* and *Style Imitation*, the creation of a visualization can be formed as generating a visualization that is similar to the collected visualization examples [28], [45], [48], [49]. In *User Profiling*,

predicting user actions can be formed as generating the next user actions, so that the action sequence is similar to the collected user action sequences [32], [66]. In *VIS Reading*, the human perception of a visualization can be modeled through latent features [67].

Among all the generative models, GAN is a hot research topic in the field of deep learning [129]. In ML4VIS, GANs have been exclusively used in *Data-VIS Mapping* for scientific visualization, including volume rendering [28], density maps [48], and in-situ visualization of ensemble simulations [45]. GANs can effectively synthesize visualizations without actually running the rendering process, which can be time-consuming and computationally expensive. However, training GANs is usually difficult due to non-convergence and model collapse [130]. Therefore, traditional (non-deep-learning-based) generative models, such as Markov model, Gaussian mixture model, variational autoencoder, are also popular, especially when the training data is small and the problem is relatively simple. For example, a click event has a relatively simple data format compared with a visualization image and thus can be generated using a less complicated ML model. Ottley et al. [66] used a hidden Markov model to generate future click events on a scatter plot visualization. The evolution of user attention on the visualization is modeled based on the collected clickstream data. The future click events are then generated based on the user attention evolution.

In **dimension reduction**, an ML model transforms the data from a high-dimensional space to a low-dimensional space. In the collected ML4VIS studies, dimension reduction is used in five visualization processes for two main purposes. In *Data Processing4VIS* and *Data-VIS Mapping*, dimension reduction is often used to process high-dimensional data for the visualization in a 2D space [19], [29]. For example, to visualize streaming data with varying dimensions, Fujiwara et al. [53] improved the incremental principal component analysis (PCA) and proposed a new dimension reduction solution. In *Style Imitation*, *VIS Reading*, and *VIS Interaction*, dimension reduction is often used to extract representative features for effective analysis [53], [56]. For example, to characterize the distribution patterns showed in scatter plots, Jo and Seo [56] trained a β-variational autoencoder to transform a scatter plot image ($64 \times 64 \times 1$) to a 32×1 feature vector. The 32 features capture the underlying data distribution of the scatter plot and can be used to predict the human-perceived distances between scatter plots.

Popular dimension reduction methods include PCA, MDS, t-SNE, UMAP, and autoencoder. PCA, MDS, t-SNE, and UMAP have been widely used for visualizing high dimensional data on the 2D space. While the liner methods such as PCA and MDS are good at preserving global structures, non-linear methods such as t-SNE and UMAP are good at preserving local structures. Autoencoders can be easily combined with other ML models to extract features from complicated inputs. For example, an autoencoder can be combined with graph neural networks (GNNs) to extract features from graphs [49], combined with 3D convolution to extract features from 3D streamlines and surfaces [51], and combined with CNNs to extract features from images [54], [56].

In **clustering**, a model divides unlabeled data into a number of groups based on their similarity. Clustering is a classical problem and has been used for many applications in the visualization field. In *Style Imitation*, collected training examples can be clustered based on their color styles [62] to guide the creation of a new visualization. In *VIS Interaction* and *VIS Reading*, visual objects in a visualization can be clustered to better understand and interact with this visualization [51], [65], [67]. Generally speaking, clustering algorithms can achieve the most satisfactory results when

the input features are representative. Therefore, when clustering complicated inputs (*e.g.*, images, natural language, extremely high-dimensional features), it is necessary to apply feature engineering or dimension reduction to extract representative features from the original data. We refer the readers to [116], [131] for more details about clustering algorithms.

6.2.4 Semi-supervised Learning

In semi-supervised learning. a model is trained using a small amount of labeled data with a large amount of unlabeled data. Semi-supervised learning is a combination between unsupervised learning (no data are labeled) and supervised learning (all data are labeled). Overall, semi-supervised learning is a less explored type of learning in ML4VIS. The only two studies [82], [83] that employed semi-supervised learning belong to the same sub-category, *i.e.*, active learning. More specifically, Luo et al. [82] employed active learning to interactively clean data in *Data Processing4VIS*. Users are asked to label whether certain data items are duplicated and will influence the visualization. Their labels will then be used to improve the classification of data duplicates. Lekschas et al. [83] applied active learning to improve the interactive visual pattern search in *VIS Interaction*. A set of unlabeled visual regions are identified using an active learning strategy, and labeled by the users based on whether they contain interesting visual patterns. A classification model is then iteratively trained using the user labels to predict the interestingness of other unlabeled regions of the visualization.

6.2.5 Reinforcement Learning

In reinforcement learning, an agent learns to take actions in an environment to maximize the cumulative rewards. Different from supervised and unsupervised learning, the task of reinforcement learning is not to directly generate end results (*e.g.*, classification labels, latent features) but to take actions in an interactive environment (*e.g.*, playing a GO game).

To be solved through reinforcement learning, a problem needs to be formalized as finding a sequence of actions that maximize the cumulative rewards. More specifically, the problem needs to be described as a Markov decision process, which contains a state space S, an action space A, a transition function, and a reward function. At each time step, the agent is in a certain state $s_t \in S$ and chooses an action $a_u \in A$. The agent moves to a new state s_{t+1} in the next time step based on the transition function $s_{t+1} = T(s_t, a_u)$ and gets a corresponding reward r based on the reward function $s_{t+1} = T(s_t, a_u)$ [132]. For example, PlotThread [75] formalizes the problem of creating a user-preferred storyline layout as performing a sequence of layout modifications on a less satisfying layout. All possible layouts form the state space and all possible layout modifications form the action space. The reward is the similarity between the current layout and the user-preferred layout. The transition is implemented by applying the layout modification to the current layout.

Reinforcement learning has made significant success in a variety of tasks and a large number of reinforcement learning models have been proposed. However, only a limited number of ML4VIS studies have used reinforcement learning, including asynchronous advantage actor-critic [133] (used in PlotThread [75]), policy gradient [134] (used in MobileVisFixer [74]), and deep Q-learning [135] (used in Table2Chart [88]). How to apply reinforcement learning to ML4VIS still requires further exploration.

7 RESEARCH CHALLENGES & OPPORTUNITIES

In this section, we discuss the research challenges and opportunities of ML4VIS that are derived from our survey, hoping to provide insights for both researchers and practitioners in visualization.

7.1 From the Aspect of ML

More Diverse Types of ML: According to our survey, semi-supervised learning and reinforcement learning are less considered in ML4VIS studies, even though they demonstrate promising properties for ML4VIS studies.

Semi-supervised learning extracts knowledge from data that only a small proportion is labeled. By combining unlabeled data with labeled data, semi-supervised learning not only reduces the expense of data labeling but also surpasses both supervised learning (trained only using the small amount of labeled data) and unsupervised learning (trained using all data without their labels). For many problems in ML4VIS (*e.g.*, generating suitable visualizations), the data labeling requires human experts (*e.g.*, skilled designers). Sometimes, it is infeasible to get a large and fully labeled training dataset. The application of semi-supervised learning can help address this issue by reducing the required number of labels. Initial exploration of semi-supervised learning in ML4VIS has been conducted. For example, Lekschas et al. [83] and Luo et al. [82] have demonstrated the effectiveness of using active learning in data cleaning and visual pattern search. However, how to use more techniques in semi-supervised learning for more visualization processes in ML4VIS still requires further investigation.

Reinforcement learning is able to learn sequential decisions without enumerating all possible training examples. While current ML4VIS studies that use reinforcement learning are relatively sparse (4/88), previous ML studies have successfully applied reinforcement learning to human-computer interaction and demonstrated the capabilities of reinforcement learning in modeling user-data interactions [136], [137] and in making personalized recommendations [138]. These previous studies in general user interaction show the potential of reinforcement learning in various visualization problems, such as predicting next-step interaction in *VIS Interaction*, modeling users in *User Profiling*, and recommending suitable visual representations in *Data-VIS Mapping* and *Insight Communication*.

Note that we summarize the collected ML4VIS studies in terms of ML tasks to better align ML capabilities with visualization needs in this survey. We also encourage the readers to consider ML from other perspectives and embrace more diverse ML techniques (*e.g.*, federated learning, transform learning).

Public High-quality Datasets: In this survey, we notice that most papers need to construct their own datasets due to the lack of public visualization datasets [35], [69], [139]. In existing ML4VIS studies, the dataset quality is often limited by the size of data and the reliability of the label, and may potentially endanger the validity of the obtained ML models. For example, DeepEye [35] learns to classify "good"/"bad" visualizations based on the training examples labeled by 100 students, whose knowledge about visualization is unclear. VizML [69] trains a visualization recommender using collected online visualizations, yet previous studies have pointed out that online visualizations have a large proportion of deceptive visualizations [140]. These studies prove the effectiveness of ML in solving visualization problems, but the questionable quality of their training data can degrade the performances of the ML models (*e.g.*, the quality of the recommended visualizations). Public and high-quality datasets are needed to further improve the application of ML in visualizations. We hope this survey can encourage more research on related directions.

Benchmark Tasks: ML4VIS studies are still at their early stage and benchmark tasks for ML4VIS remain unclear. However, benchmark tasks are important for the progress of ML techniques [116]. In the field of ML, ML models are commonly evaluated by comparing their performances with the state-of-the-art on benchmark tasks (*e.g.*, question answering, instance segmentation). Advanced tasks can be accomplished by combining the methods developed for multiple benchmark tasks. The lack of benchmark tasks in ML4VIS makes peer comparison difficult and can lead to negative impacts on follow-up studies.

Meanwhile, we find that existing ML4VIS studies have intensively investigated several tasks, including graphic elements extraction [21], [60], visualizations generation [35], [41], [69], and graph layout [17], [49], [50]. These tasks can be the start point to define benchmark tasks in ML4VIS.

7.2 From the Aspect of VIS

Diversity in Visualization We notice that the forms of visualization are limited in the collected papers. First, current ML4VIS studies exclusively focus on static visualizations. Even though other forms of visualizations, such as animated transitions and data videos, are popular and effective in information communication, these visualizations are rarely discussed in the collected ML4VIS studies. We conjecture this is partly caused by the difficulty of understanding and generating dynamic visualizations using ML techniques. Take image-format training data as an example. Compared with statistic visualization (images), dynamic visualizations (videos) require a larger set of training data, more complicated ML models, and more powerful computational resources. Second, most ML4VIS studies focus on certain types of standard visualizations, especially bar charts, scatter plots, and graphs. Other types of visualizations, such as treemap, streamgraph, and parallel coordinates, are not discussed despite their wide popularity in the real world. Moreover, chart decorations are overlooked even though they are important for the memorability and user engagement of visualization. Multi-view visualizations are rarely mentioned even though they are commonly used in visual analysis. Since the success of visualization depends on choosing proper forms of visualization [141], we believe good opportunities exist in applying ML techniques to more diverse visualizations.

Towards VIS-Tailored ML: In this survey, most ML4VIS studies directly apply ML techniques developed by ML researchers. However, general ML techniques do not always suit well for the specific problems in visualization. Take computer vision in *VIS Reading* as an example. General ML techniques are mainly developed for natural images and cannot be directly applied to images with charts and diagrams [13], [60]. Chen et al. [60] employed a deep learning model to understand timeline infographics, but found that they needed a series of post-processing to address the poor model performance caused by the difference between natural images and charts. The authors also highlighted the importance to develop ML techniques specified for visualizations. We hope to see more novel ML techniques that are tailored for the unique needs in visualizations, such as the differences between chart images and natural images [13], [60], the differences between user interactions with visualizations and general user interactions [120].

Human-Machine Collaboration: Most ML4VIS studies treat ML as a black box without the involvement of humans. Take *Data-VIS Mapping* as an example. ML directly generates suitable visualizations from the given data in an end-to-end manner [41], [69]. However, visualization is by nature a human-centric field: visual design often relies on the creativity of designers; visual analysis usually depends on the knowledge and experience of domain experts. Keeping humans in the loop is crucial for the success of visualization. More importantly, a perfect ML model can rarely be obtained limited by the quality of training data and the ambiguity of the problem. Unsatisfactory model results will decrease user trust in ML4VIS and hinder the wide adoption of ML4VIS. Take *VIS Interaction* as an example, automated interaction refinement can confuse users when the prediction is at odds with the user's expectation [36], [46].

Considering the possibility of unsatisfactory model results, many ML4VIS studies allow user participation by supporting post-hoc refinement, such as adjusting the automatically selected nodes [46] and modifying the auto-generated visualizations [59]. Recently, initial exploration in human-machine collaboration has been conducted [75], [83]. For example, PlotThread [75] enables the ML agent and the human designer to work in the same canvas and modify the layout of a storyline collaboratively. However, it requires further exploration of how to support a close human-machine collaboration where human users are able to interpret, modify, and improve the ML. We envision that a close human-machine collaboration in ML4VIS can be achieved through explainable and interactive ML, which not only increases user trust by providing transparent predictions [94] but also improves ML performances by utilizing user feedback [142].

Towards User-friendly ML4VIS: The employment of ML not only provides opportunities but also poses new challenges in designing visualizations. For example, using ML to automatically refine user interaction or generate visualizations can violate the design principle of *"minimize unexpected changes"* [99], [143].

Some ML4VIS studies have discussed the usability issues of ML4VIS systems and proposed design suggestions in discussion and future work. For example, Wang et al. [50] suggested that self-exploring should be supported for expert users in automated graph drawing. When evaluating their proposed natural language interface for visual analysis, Yu et al. [44] found that users were often confused about why their queries were rejected and did not know how to modify the queries. Yu et al. suggested that possible corrections of rejected queries should be provided to improve the usability of the system.

While these design suggestions are insightful, they are scattered among different papers. A systematic summary of these design guidelines is still missing. More importantly, systematic cognitive studies are required to help designers better understand user behaviors and expectations in this new ML4VIS scenario.

8 DISCUSSION

In this section, we discuss the restrictions of ML4VIS and the limitations of this survey. The discussion of ML restrictions complements the review of ML4VIS, aiming to provide visualization researchers and practitioners a comprehensive introduction of ML4VIS. The limitations of this survey highlight several promising directions for future surveys to further deepen our understanding about ML4VIS.

8.1 ML is Not a Panacea

In spite of the great success of ML4VIS, we still want to emphasize that ML is not the only technical solution towards intelligent visualization. ML-based methods are good at recognizing complicated patterns from a huge amount of data. However, when the pattern is simple or when training data is hard to obtain, ML may not be the optimal solution. Previous studies [144], [145] have demonstrated the effectiveness of non-ML-based methods, such as expert-defined rules and statistic summaries, in solving visualization problems, especially when the training data is hard to collect, when the problem space is sparse, or when the solution can be easily described through rules. For instance, ML techniques are widely used to understand visualizations in the form of bitmap images [3], [60]. But if the given input is an SVG file, a traverse of the SVG tree can effectively deconstruct the visualization and extract the content [146], [147], saving the great efforts to collect training data and the high computation cost to train an ML model. Another example is FlowSense [44], which uses a combination of pre-defined rules and ML methods to capture natural language input patterns due to the lack of labeled examples and computational resources. While we hope this survey can demonstrate the effectiveness of ML in visualizations and promote more ML4VIS studies, visualization researchers should also pay attention to the unique advantages of non-ML-based methods and apply them when applicable.

8.2 Limitations & Future Work

This survey comes with certain limitations due to the adopted approaches. First, we only investigate technique & application papers in this survey. Apart from technique & application papers, evaluation papers are also important for the field of ML4VIS. For example, Hearst et al. [148] conducted a crowdsourcing study with 274 participants to understand how to design natural language interfaces for visual analytics systems. A review of these evaluation papers can help us understand how users perceive ML4VIS applications and guide the design of ML4VIS. Second, we summarize the main purposes of employing ML in different visualization processes based on our analysis of the collected papers. We believe these analysis can provide useful insights. However, it is possible that there are other under-explored areas of employing ML. ML4VIS is still an ongoing research field, we expect more studies to be conducted to expand our understanding about it.

9 CONCLUSION

In this paper, we survey 88 papers to understand the current practices in ML4VIS research, *i.e.*, employing ML techniques for solving problems related to data visualization. Guided by two motivating questions: *"what visualization processes are assisted by ML?"* and *"what ML capabilities are used for visualization?"*, we summarize seven main visualization processes that are benefiting from the application of ML techniques. The seven visualization processes are also aligned with the learning tasks in ML to reveal how the needs in visualization can be formed as ML tasks. An ML4VIS pipeline is also proposed to organize the seven visualization processes and map out the role of ML4VIS in general visualizations. We further discuss the current practices and future research opportunities in ML4VIS based on our analysis of the collected studies. We believe this survey can provide useful insights into the field of ML4VIS and promote future studies.

ACKNOWLEDGMENTS

REFERENCES

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

[2] D. Gotz and Z. Wen, "Behavior-driven visualization recommendation," in *Proceedings of the International Conference on Intelligent User Interfaces*, 2009, pp. 315–324.

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

[4] A. Key, B. Howe, D. Perry, and C. Aragon, "Vizdeck: Self-organizing dashboards for visual analytics," in *Proceedings of the ACM International Conference on Management of Data*, ser. SIGMOD '12. New York, NY, USA: ACM, 2012, p. 681–684.

[5] B. Steichen, G. Carenini, and C. Conati, "User-adaptive information visualization: using eye gaze data to infer visualization tasks and user cognitive abilities," in *Proceedings of the International Conference on Intelligent User Interfaces*, 2013, pp. 317–328.

[6] E. T. Brown, A. Ottley, H. Zhao, Q. Lin, R. Souvenir, A. Endert, and R. Chang, "Finding waldo: Learning about users from their interactions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1663–1672, 2014.

[7] S. Lallé, D. Toker, C. Conati, and G. Carenini, "Prediction of users' learning curves for adaptation while using an information visualization," in *IUI*. ACM, 2015, pp. 357–368.

[8] D. Toker, B. Steichen, M. Gingerich, C. Conati, and G. Carenini, "Towards facilitating user skill acquisition: Identifying untrained visualization users through eye tracking," in *Proceedings of the International Conference on Intelligent User Interfaces*, ser. IUI '14. New York, NY, USA: ACM, 2014, p. 105–114.

[9] M. Sedlmair and M. Aupetit, "Data-driven evaluation of visual quality measures," *Computer Graphics Forum*, vol. 34, no. 3, p. 201–210, Jun. 2015.

[10] B. Mutlu, E. Veas, and C. Trattner, "Vizrec: Recommending personalized visualizations," *ACM Trans. Interact. Intell. Syst.*, vol. 6, no. 4, Nov. 2016.

[11] M. Aupetit and M. Sedlmair, "Sepme: 2002 new visual separation measures," in *Pacific Visualization Symposium*. IEEE, 2016, pp. 1–8.

[12] N. Siegel, Z. Horvitz, R. Levin, S. Divvala, and A. Farhadi, "Figureseer: Parsing result-figures in research papers," in *European Conference on Computer Vision*. Springer, 2016, pp. 664–680.

[13] A. Kembhavi, M. Salvato, E. Kolve, M. J. Seo, H. Hajishirzi, and A. Farhadi, "A diagram is worth a dozen images," in *Proceedings of the 14th European Conference on Computer Vision (ECCV)*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9908. Springer, 2016, pp. 235–251.

[14] R. Al-Zaidy, S. Choudhury, and C. Giles, "Automatic summary generation for scientific data charts," in *Proceedings of the Conference on Artificial Intelligence*, vol. WS-16-01 - WS-16-15. AI Access Foundation, Jan. 2016, pp. 658–663, proceedings of the Conference on Artificial Intelligence.

[15] N. Pezzotti, B. P. Lelieveldt, L. Van Der Maaten, T. Höllt, E. Eisemann, and A. Vilanova, "Approximated and user steerable tsne for progressive visual analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 7, pp. 1739–1752, 2016.

[16] J. Poco, A. Mayhua, and J. Heer, "Extracting and retargeting color mappings from bitmap images of visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 637–646, 2017.

[17] O.-H. Kwon, T. Crnovrsanin, and K.-L. Ma, "What would a graph look like in this layout? a machine learning approach to large graph visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 478–488, 2017.

[18] 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 *Proceedings of the ACM Symposium on User Interface Software and Technology*, 2017, pp. 57–69.

[19] D. K. Saha, V. D. Calhoun, S. R. Panta, and S. M. Plis, "See without looking: Joint visualization of sensitive multi-site datasets," in *26th International Joint Conference on Artificial Intelligence, IJCAI 2017*. International Joint Conferences on Artificial Intelligence, 2017, pp. 2672–2678.

[20] J. F. Kruiger, P. E. Rauber, R. M. Martins, A. Kerren, S. G. Kobourov, and A. Telea, "Graph layouts by t-sne," *Computer Graphics Forum*, vol. 36, no. 3, pp. 283–294, 2017.

[21] J. Poco and J. Heer, "Reverse-engineering visualizations: Recovering visual encodings from chart images," *Computer Graphics Forum*, vol. 36, no. 3, pp. 353–363, 2017.

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

[23] Z. Bylinskii, S. Alsheikh, S. Madan, A. Recasens, K. Zhong, H. Pfister, F. Durand, and A. Oliva, "Understanding infographics through textual and visual tag prediction," *arXiv preprint arXiv:1709.09215*, 2017.

[24] R. A. Al-Zaidy and C. L. Giles, "A machine learning approach for semantic structuring of scientific charts in scholarly documents," in *Proceedings of the Conference on Artificial Intelligence*, S. P. Singh and S. Markovitch, Eds. AAAI Press, 2017, pp. 4644–4649.

[25] T. Siddiqui, P. Luh, Z. Wang, K. Karahalios, and A. Parameswaran, "Shapesearch: flexible pattern-based querying of trend line visualizations," *Proceedings of the VLDB Endowment*, vol. 11, no. 12, pp. 1962–1965, 2018.

[26] C. C. Gramazio, J. Huang, and D. H. Laidlaw, "An analysis of automated visual analysis classification: Interactive visualization task inference of cancer genomics domain experts," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 8, pp. 2270–2283, 2017.

[27] 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 Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 438–448, 2019.

[28] M. Berger, J. Li, and J. A. Levine, "A generative model for volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 4, pp. 1636–1650, 2019.

[29] Y. Wang, K. Feng, X. Chu, J. Zhang, C.-W. Fu, M. Sedlmair, X. Yu, and B. Chen, "A perception-driven approach to supervised dimensionality reduction for visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 5, pp. 1828–1840, 2017.

[30] D. Haehn, J. Tompkin, and H. Pfister, "Evaluating 'graphical perception' with cnns," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 641–650, 2019.

[31] Y. Luo, X. Qin, N. Tang, G. Li, and X. Wang, "Deepeye: Creating good data visualizations by keyword search," in *Proceedings of the ACM International Conference on Management of Data*, 2018, pp. 1733–1736.

[32] T. Milo and A. Somech, "Next-step suggestions for modern interactive data analysis platforms," in *Proceedings of the ACM International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 576–585.

[33] T. Zhou, H. Qian, Z. Shen, C. Zhang, C. Wang, S. Liu, and W. Ou, "JUMP: a jointly predictor for user click and dwell time," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, J. Lang, Ed. ijcai.org, 2018, pp. 3704–3710.

[34] S. E. Kahou, V. Michalski, A. Atkinson, Á. Kádár, A. Trischler, and Y. Bengio, "FigureQA: An annotated figure dataset for visual reasoning," in *International Conference on Learning Representations, Workshop Track Proceedings*. OpenReview.net, 2018.

[35] Y. Luo, X. Qin, N. Tang, and G. Li, "Deepeye: towards automatic data visualization," in *Proceedings of the IEEE International Conference on Data Engineering*. IEEE, 2018, pp. 101–112.

[36] C. Fan and H. Hauser, "Fast and accurate cnn-based brushing in scatterplots," *Computer Graphics Forum*, vol. 37, no. 3, pp. 111–120, 2018.

[37] M. Chegini, L. Shao, R. Gregor, D. J. Lehmann, K. Andrews, and T. Schreck, "Interactive visual exploration of local patterns in large scatterplot spaces," *Computer Graphics Forum*, vol. 37, no. 3, pp. 99–109, 2018.

[38] K. Kafle, B. Price, S. Cohen, and C. Kanan, "DVQA: Understanding data visualizations via question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5648–5656.

[39] D. Kim, Y. Yoo, J.-S. Kim, S. Lee, and N. Kwak, "Dynamic graph generation network: Generating relational knowledge from diagrams," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4167–4175.

[40] L. Battle, P. Duan, Z. Miranda, D. Mukusheva, R. Chang, and M. Stonebraker, "Beagle: Automated extraction and interpretation of visualizations from the web," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2018, pp. 1–8.

[41] V. Dibia and C. Demiralp, "Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks," *IEEE Computer Graphics and Applications*, vol. 39, no. 5, pp. 33–46, 2019.

[42] H. Haleem, Y. Wang, A. Puri, S. Wadhwa, and H. Qu, "Evaluating the readability of force directed graph layouts: A deep learning approach," *IEEE Computer Graphics and Applications*, vol. 39, no. 4, pp. 40–53, 2019.

[43] S. Madan, Z. Bylinskii, M. Tancik, A. Recasens, K. Zhong, S. Alsheikh, H. Pfister, A. Oliva, and F. Durand, "Synthetically trained icon proposals for parsing and summarizing infographics," *arXiv preprint arXiv:1807.10441*, 2018.

[44] B. Yu and C. T. Silva, "Flowsense: A natural language interface for visual data exploration within a dataflow system," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 1–11, 2019.

[45] W. He, J. Wang, H. Guo, K.-C. Wang, H.-W. Shen, M. Raj, Y. S. Nashed, and T. Peterka, "InSituNet: Deep image synthesis for parameter space exploration of ensemble simulations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 23–33, 2019.

[46] Z. Chen, W. Zeng, Z. Yang, L. Yu, C.-W. Fu, and H. Qu, "LassoNet: Deep lasso-selection of 3d point clouds," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 195–204, 2019.

[47] J. Han and C. Wang, "TSR-TVD: Temporal super-resolution for time-varying data analysis and visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 205–215, 2019.

[48] C. Chen, C. Wang, X. Bai, P. Zhang, and C. Li, "Generativemap: Visualization and exploration of dynamic density maps via generative learning model," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 216–226, 2019.

[49] O.-H. Kwon and K.-L. Ma, "A deep generative model for graph layout," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 665–675, 2019.

[50] Y. Wang, Z. Jin, Q. Wang, W. Cui, T. Ma, and H. Qu, "Deepdrawing: A deep learning approach to graph drawing," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 676–686, 2019.

[51] J. Han, J. Tao, and C. Wang, "Flownet: A deep learning framework for clustering and selection of streamlines and stream surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 4, pp. 1732–1744, 2020.

[52] E. Wall, A. Arcalgud, K. Gupta, and A. Jo, "A markov model of users' interactive behavior in scatterplots," in *Proceedings of the 2019 IEEE Visualization Conference*, 2019, pp. 81–85.

[53] T. Fujiwara, J.-K. Chou, S. Shilpika, P. Xu, L. Ren, and K.-L. Ma, "An incremental dimensionality reduction method for visualizing streaming multidimensional data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 418–428, 2019.

[54] X. Fu, Y. Wang, H. Dong, W. Cui, and H. Zhang, "Visualization assessment: A machine learning approach," in *Proceedings of the IEEE Visualization Conference*. IEEE, 2019, pp. 126–130.

[55] W. P. Porter, Y. Xing, B. R. von Ohlen, J. Han, and C. Wang, "A deep learning approach to selecting representative time steps for time-varying multivariate data," in *Proceedings of the IEEE Visualization Conference*. IEEE, 2019, pp. 1–5.

[56] J. Jo and J. Seo, "Disentangled representation of data distributions in scatterplots," in *Proceedings of the IEEE Visualization Conference*. IEEE, 2019, pp. 136–140.

[57] Y. Ma, A. K. Tung, W. Wang, X. Gao, Z. Pan, and W. Chen, "ScatterNet: A deep subjective similarity model for visual analysis of scatterplots," *IEEE Transactions on Visualization and Computer Graphics*, 2018.

[58] 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 Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 895–905, 2019.

[59] 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 Transactions on Visualization and Computer graphics*, vol. 26, no. 1, pp. 906–916, 2019.

[60] Z. Chen, Y. Wang, Q. Wang, Y. Wang, and H. Qu, "Towards automated infographic design: Deep learning-based auto-extraction of extensible timeline," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 917–926, 2019.

[61] Y. Wang, Z. Zhong, and J. Hua, "Deeporgannet: On-the-fly reconstruction and visualization of 3d/4d lung models from single-view projections by deep deformation network," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 960–970, 2019.

[62] S. Smart, K. Wu, and D. A. Szafir, "Color crafting: Automating the construction of designer quality color ramps," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 1215–1225, Jan 2020.

[63] Z. Huang, Y. Zhao, W. Chen, S. Gao, K. Yu, W. Xu, M. Tang, M. Zhu, and M. Xu, "A natural-language-based visual query approach of uncertain human trajectories," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 1256–1266, 2019.

[64] F. Hong, C. Liu, and X. Yuan, "Dnn-volvis: Interactive volume visualization supported by deep neural network," in *Pacific Visualization Symposium*. IEEE, 2019, pp. 282–291.

[65] C. Fan and H. Hauser, "On KDE-based Brushing in Scatterplots and how it Compares to CNN-based Brushing," in *Machine Learning Methods in Visualisation for Big Data*, D. Archambault, I. Nabney, and J. Peltonen, Eds. The Eurographics Association, 2019.

[66] A. Ottley, R. Garnett, and R. Wan, "Follow the clicks: Learning and anticipating mouse interactions during exploratory data analysis," *Computer Graphics Forum*, vol. 38, no. 3, pp. 41–52, 2019.

[67] M. M. Abbas, M. Aupetit, M. Sedlmair, and H. Bensmail, "Clustme: A visual quality measure for ranking monochrome scatterplots based on cluster patterns," *Computer Graphics Forum*, vol. 38, no. 3, pp. 225–236, 2019.

[68] J. Kassel and M. Rohs, "Online learning of visualization preferences through dueling bandits for enhancing visualization recommendations," in *EuroVis (Short Papers)*. Eurographics Association, 2019, pp. 85–89.

[69] K. Hu, M. A. Bakker, S. Li, T. Kraska, and C. Hidalgo, "VizML: A machine learning approach to visualization recommendation," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, ser. CHI '19, 2019.

[70] C. Fan and H. Hauser, "Personalized sketch-based brushing in scatterplots," *IEEE computer graphics and applications*, vol. 39, no. 4, pp. 28–39, 2019.

[71] K. Kafle, R. Shrestha, S. Cohen, B. Price, and C. Kanan, "Answering questions about data visualizations using efficient bimodal fusion," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 1498–1507.

[72] H. Mohammed, "Continuous prefetch for interactive data applications," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2020, pp. 2841–2843.

[73] P. Zhang, C. Li, and C. Wang, "Viscode: Embedding information in visualization images using encoder-decoder network," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[74] A. Wu, W. Tong, T. Dwyer, B. Lee, P. Isenberg, and H. Qu, "MobileVisFixer: Tailoring web visualizations for mobile phones leveraging an explainable reinforcement learning framework," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[75] T. Tang, R. Li, X. Wu, S. Liu, J. Knittel, S. Koch, T. Ertl, L. Yu, P. Ren, and Y. Wu, "PlotThread: Creating expressive storyline visualizations using reinforcement learning," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[76] 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 Transactions on Visualization and Computer Graphics*, 2020.

[77] Y. Wang, G. Yan, H. Zhu, S. Buch, Y. Wang, E. M. Haacke, J. Hua, and Z. Zhong, "Vc-net: Deep volume-composition networks for segmentation and visualization of highly sparse and noisy image data," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[78] M. Oppermann, R. Kincaid, and T. Munzner, "Vizcommender: Computing text-based similarity in visualization repositories for content-based recommendations," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[79] C. Fosco, V. Casser, A. K. Bedi, P. O'Donovan, A. Hertzmann, and Z. Bylinskii, "Predicting visual importance across graphic design types," in *Proceedings of the ACM Symposium on User Interface Software and Technology*, 2020, pp. 249–260.

[80] L. Giovannangeli, R. Bourqui, R. Giot, and D. Auber, "Toward automatic comparison of visualization techniques: Application to graph visualization," *Visual Informatics*, 2020.

[81] C. Liu, L. Xie, Y. Han, X. Yuan *et al.*, "AutoCaption: An approach to generate natural language description from visualization automatically," in *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 2020, pp. 191–195.

[82] Y. Luo, C. Chai, X. Qin, N. Tang, and G. Li, "Interactive cleaning for progressive visualization through composite questions," in *Proceedings of the IEEE International Conference on Data Engineering*. IEEE, 2020.

[83] F. Lekschas, B. Peterson, D. Haehn, E. Ma, N. Gehlenborg, and H. Pfister, "Peax: Interactive visual pattern search in sequential data using unsupervised deep representation learning," *Computer Graphics Forum*, vol. 39, no. 3, pp. 167–179, 2020.

[84] N. Zhao, N. W. Kim, L. M. Herman, H. Pfister, R. W. Lau, J. Echevarria, and Z. Bylinskii, "ICONATE: Automatic compound icon generation and ideation," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2020, pp. 1–13.

[85] C. Lai, Z. Lin, R. Jiang, Y. Han, C. Liu, and X. Yuan, "Automatic annotation synchronizing with textual description for visualization," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2020, pp. 1–13.

[86] D. H. Kim, E. Hoque, and M. Agrawala, "Answering questions about charts and generating visual explanations," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2020, pp. 1–13.

[87] M. Lu, C. Wang, J. Lanir, N. Zhao, H. Pfister, D. Cohen-Or, and H. Huang, "Exploring visual information flows in infographics," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2020, pp. 1–12.

[88] M. Zhou, Q. Li, Y. Li, S. Han, and D. Zhang, "Table2charts: Learning shared representations for recommending charts on multi-dimensional data," *arXiv preprint arXiv:2008.11015*, 2020.

[89] J. Mackinlay, "Automating the design of graphical presentations of relational information," *Acm Transactions On Graphics (Tog)*, vol. 5, no. 2, pp. 110–141, 1986.

[90] D. A. Keim, T. Munzner, F. Rossi, and M. Verleysen, "Bridging information visualization with machine learning (dagstuhl seminar 15101)," in *Dagstuhl Reports*, vol. 5, no. 3. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

[91] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon, "Visual analytics: Definition, process, and challenges," in *Information Visualization*. Springer, 2008, pp. 154–175.

[92] A. Endert, W. Ribarsky, C. Turkay, B. W. Wong, I. Nabney, I. D. Blanco, and F. Rossi, "The state of the art in integrating machine learning into visual analytics," *Computer Graphics Forum*, vol. 36, no. 8, pp. 458–486, 2017.

[93] J. Yuan, C. Chen, W. Yang, M. Liu, J. Xia, and S. Liu, "A survey of visual analytics techniques for machine learning," *Computational Visual Media*, pp. 1–34, 2020.

[94] A. Chatzimparmpas, R. M. Martins, I. Jusufi, K. Kucher, F. Rossi, and A. Kerren, "The state of the art in enhancing trust in machine learning models with the use of visualizations," in *Computer Graphics Forum*, vol. 39, no. 3. Wiley Online Library, 2020, pp. 713–756.

[95] A. Chatzimparmpas, R. M. Martins, I. Jusufi, and A. Kerren, "A survey of surveys on the use of visualization for interpreting machine learning models," *Information Visualization*, vol. 19, no. 3, pp. 207–233, 2020.

[96] S. Zhu, G. Sun, Q. Jiang, M. Zha, and R. Liang, "A survey on automatic infographics and visualization recommendations," *Visual Informatics*, vol. 4, no. 3, pp. 24–40, 2020.

[97] M. Vartak, S. Huang, T. Siddiqui, S. Madden, and A. Parameswaran, "Towards visualization recommendation systems," *ACM SIGMOD Record*, vol. 45, no. 4, pp. 34–39, 2017.

[98] Q. Yang, N. Banovic, and J. Zimmerman, "Mapping machine learning advances from hci research to reveal starting places for design innovation," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2018, pp. 1–11.

[99] S. Amershi, D. S. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. T. Iqbal, P. N. Bennett, K. Inkpen, J. Teevan, R. KikinGil, and E. Horvitz, "Guidelines for human-ai interaction," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, 2019, p. 3.

[100] E. Horvitz, "Principles of mixed-initiative user interfaces," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, M. G. Williams and M. W. Altom, Eds. ACM, 1999, pp. 159–166.

[101] B. Saket, D. Moritz, H. Lin, V. Dibia, C. Demiralp, and J. Heer, "Beyond heuristics: Learning visualization design," *arXiv preprint arXiv:1807.06641*, 2018.

[102] T. Munzner, "A nested model for visualization design and validation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 921–928, 2009.

[103] J. J. Van Wijk, "The value of visualization," in *Proceedings of the IEEE Visualization Conference*. IEEE, 2005, pp. 79–86.

[104] M. Card, *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[105] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim, "Visual interaction with dimensionality reduction: A structured literature analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 241–250, 2016.

[106] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. P. Ellis, and D. A. Keim, "Knowledge generation model for visual analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1604–1613, 2014.

[107] M. El-Assady, F. Sperrle, O. Deussen, D. Keim, and C. Collins, "Visual analytics for topic model optimization based on user-steerable speculative execution," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 374–384, 2018.

[108] D. Sacha, M. Kraus, D. A. Keim, and M. Chen, "VIS4ML: an ontology for visual analytics assisted machine learning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 385–395, 2019.

[109] T. Spinner, U. Schlegel, H. Schäfer, and M. El-Assady, "explainer: A visual analytics framework for interactive and explainable machine learning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 1064–1074, 2019.

[110] S. Liu, X. Wang, C. Collins, W. Dou, F. Ou-Yang, M. El-Assady, L. Jiang, and D. A. Keim, "Bridging text visualization and mining: A task-driven survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 7, pp. 2482–2504, 2019.

[111] G. Scholar, "Google scholar top publications," https://scholar.google.com/citations?view_op=top_venues.

[112] 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 *Proceedings of the ACM Conference on Human Factors in Computing Systems*, ser. CHI '17, 2017, p. 2648–2659.

[113] S. Hazarika, H. Li, K.-C. Wang, H.-W. Shen, and C.-S. Chou, "NNVA: Neural network assisted visual analysis of yeast cell polarization simulation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 34–44, 2019.

[114] Y. Guo, S. Guo, Z. Jin, S. Kaul, D. Gotz, and N. Cao, "Survey on visual analysis of event sequence data," *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[115] P. R. Nicolas, *Scala for machine learning*. Packt Publishing Ltd, 2015.

[116] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[117] C. North, "Toward measuring visualization insight," *IEEE Computer Graphics and Applications*, vol. 26, no. 3, pp. 6–9, 2006.

[118] R. Chang, C. Ziemkiewicz, T. M. Green, and W. Ribarsky, "Defining insight for visual analytics," *IEEE Computer Graphics and Applications*, vol. 29, no. 2, pp. 14–17, 2009.

[119] N. W. Kim, "From exploration to explanation: Designing for visual data storytelling," Ph.D. dissertation, 2019.

[120] E. Dimara and C. Perin, "What is interaction for data visualization?" *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 119–129, 2020.

[121] A. Valli, "The design of natural interaction," *Multimedia Tools and Applications*, vol. 38, no. 3, pp. 295–305, 2008.

[122] L. Cooke, "Is the mouse a" poor man's eye tracker"?" in *Annual Conference-Society for Technical Communication*, vol. 53, 2006, p. 252.

[123] J. Fu, B. Zhu, W. Cui, S. Ge, Y. Wang, H. Zhang, R. Huang, Y. Tang, D. Zhang, and X. Ma, "Chartem: Reviving chart images with data embedding," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[124] M. A. Borkin, A. A. Vo, Z. Bylinskii, P. Isola, S. Sunkavalli, A. Oliva, and H. Pfister, "What makes a visualization memorable?" *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2306–2315, 2013.

[125] E. Segel and J. Heer, "Narrative visualization: Telling stories with data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1139–1148, 2010.

[126] K. Chowdhary, "Natural language processing," in *Fundamentals of Artificial Intelligence*. Springer, 2020, pp. 603–649.

[127] T. F. Liu, M. Craft, J. Situ, E. Yumer, R. Mech, and R. Kumar, "Learning design semantics for mobile apps," in *Proceedings of the ACM Symposium on User Interface Software and Technology*, ser. UIST '18. ACM, 2018, p. 569–579.

[128] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[129] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," *arXiv preprint arXiv:2001.06937*, 2020.

[130] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[131] G. Gan, C. Ma, and J. Wu, *Data clustering: theory, algorithms, and applications*. SIAM, 2020.

[132] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[133] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, 2016, pp. 1928–1937.

[134] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems*, 2000, pp. 1057–1063.

[135] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[136] B. McCamish, V. Ghadakchi, A. Termehchy, B. Touri, and L. Huang, "The data interaction game," in *Proceedings of the ACM International Conference on Management of Data*. ACM, 2018, p. 83–98.

[137] A. Grotov and M. de Rijke, "Online learning to rank for information retrieval: Sigir 2016 tutorial," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 1215–1218.

[138] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li, "Drn: A deep reinforcement learning framework for news recommendation," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 167–176.

[139] K. Hu, S. Gaikwad, M. Hulsebos, M. A. Bakker, E. Zgraggen, C. Hidalgo, T. Kraska, G. Li, A. Satyanarayan, and Ç. Demiralp, "Viznet: Towards a large-scale visualization learning and benchmarking repository," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–12.

[140] M. Correll and J. Heer, "Black hat visualization," in *Workshop on Dealing with Cognitive Biases in Visualisations (DECISIVe), IEEE VIS*, 2017.

[141] T. Munzner, *Visualization analysis and design*. CRC press, 2014.

[142] Y. Ming, P. Xu, H. Qu, and L. Ren, "Interpretable and steerable sequence learning via prototypes," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19. ACM, 2019, pp. 903–913.

[143] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 1990, pp. 249–256.

[144] R. Savvides, A. Henelius, E. Oikarinen, and K. Puolamäki, "Significance of patterns in data visualisations," in *Proceedings of the ACM International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1509–1517.

[145] Z. Zhao, E. Zgraggen, L. De Stefani, C. Binnig, E. Upfal, and T. Kraska, "Safe visual data exploration," in *Proceedings of the ACM International Conference on Management of Data*, 2017, pp. 1671–1674.

[146] J. Harper and M. Agrawala, "Deconstructing and restyling d3 visualizations," in *Proceedings of the ACM Symposium on User Interface Software and Technology*, 2014, pp. 253–262.

[147] ——, "Converting basic d3 charts into reusable style templates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 3, pp. 1274–1286, 2017.

[148] M. Hearst, M. Tory, and V. Setlur, "Toward interface defaults for vague modifiers in natural language interfaces for visual analysis," in *Proceedings of the IEEE Visualization Conference*. IEEE, 2019, pp. 21–25.

Qianwen Wang is a PostDoc researcher at Harvard University. Her research interests include visual analytics, explainable machine learning, and narrative visualization. Her work strives to facilitate the communication between humans and machine learning models through interactive visualization. She obtained her Ph.D in Hong Kong University of Science and Technology and her BS degree from Xi'an Jiaotong University. Please refer to http://wangqianwen0418.github.io for more details.
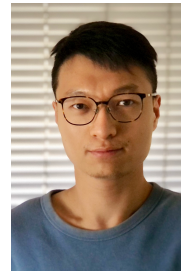
Zhutian Chen is a PostDoc researcher at University of California San Diego. His interests are in Information Visualization, Human-Computer Interaction, and Augmented Reality. He obtained his Ph.D. in Computer Science from Hong Kong University of Science and Technology in 2020. More details can be found at https://chenzhutian.org.

Yong Wang is currently an assistant professor in School of Information Systems at Singapore Management University. His research interests include data visualization, visual analytics and explainable machine learning. He obtained his Ph.D. in Computer Science from Hong Kong University of Science and Technology in 2018. He received his B.E. and M.E. from Harbin Institute of Technology and Huazhong University of Science and Technology, respectively. For more details, please refer to http://yong-wang.org.

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).