Reviving Static Charts Into Live Charts

Lu Ying, Yun Wang, Haotian Li, Shuguang Dou, Haidong Zhang, Xinyang Jiang, Huamin Qu, *Member, IEEE*, and Yingcai Wu, *Senior Member, IEEE*

***Abstract*—Data charts are prevalent across various fields due to their efficacy in conveying complex data relationships. However, static charts may sometimes struggle to engage readers and efficiently present intricate information, potentially resulting in limited understanding. We introduce "Live Charts," a new format of presentation that decomposes complex information within a chart and explains the information pieces sequentially through rich animations and accompanying audio narration. We propose an automated approach to revive static charts into Live Charts. Our method integrates GNN-based techniques to analyze the chart components and extract data from charts. Then we adopt large natural language models to generate appropriate animated visuals along with a voice-over to produce Live Charts from static ones. We conducted a thorough evaluation of our approach, which involved the model performance, use cases, a crowd-sourced user study, and expert interviews. The results demonstrate Live Charts offer a multi-sensory experience where readers can follow the information and understand the data insights better. We analyze the benefits and drawbacks of Live Charts over static charts as a new information consumption experience.**

***Index Terms*—Automatic visualization, charts, machine learning, storytelling.**

I. Introduction

STATISTICAL charts are widely used to deliver visually appealing and informative presentations of data. They have been used in various fields, such as urban [1], education [2], and science [3], to communicate data patterns [4] and insights.

However, static charts have limitations due to their unchanging nature. The absence of a clear reading order can confuse as all information is presented simultaneously, and users do not know where to start or how to locate the corresponding elements [5]. Further, static charts have a limited capacity for conveying information [6]. A chart containing excessive visual components or encodings may overwhelm users, leading to a visual burden. Moreover, static charts may not be engaging enough to maintain users' interest [7], reducing their effectiveness as a communication tool. These phenomena emphasize the need for strategies to enhance data communication via charts.

There have been several studies aimed at enhancing static charts with various solutions. One approach is to break down the complex original chart information and introduce pieces of information gradually, making it easier for users to comprehend [8]. For instance, recent research [9], [10] hierarchically decomposes a visualization or infographic design and introduces the compositions progressively. Alternatively, some studies have attempted to augment static charts' expressiveness by including available information, such as adding question-answering [11], annotations [12] or captions [13], [14].

In this paper, we introduce Live Charts, a new format that revitalizes static charts by integrating the advantages of simplifying complex information and incorporating explanatory details. Our approach employs a multi-sensory technique, including audio as an additional source of information for users. Live Charts present data sequentially, accompanied by audio narration that introduces visual encodings and elucidates the messages conveyed by the charts. For instance, when examining a static bar chart such as the one depicted in Fig. 1(a3), a Live Chart presentation would involve increasing the height of the red bars synchronously while narrating stockpile data about the United States. Adopting this format for data visualization, we explore how Live Charts may enable users to comprehend the presented information more easily and accurately.

Nevertheless, if manual creation is required for Live Charts, this new format of presentation may be deemed less useful due to the numerous steps involved, such as manually extracting data and visual encodings, designing animations, and crafting a chart narration that synchronizes with the animation. To address this concern, we further explore methods to automate the creation of Live Charts from static ones. We break down the procedure into a series of computational methods that can be automated, thereby enhancing the efficiency of this new format. We trained a Graph Neural Network (GNN) model to recover the embedded data and visual encodings of static charts on our own generated chart dataset. Subsequently, we aim to create the audio

Manuscript received 6 September 2023; revised 26 December 2023; accepted 23 April 2024. Date of publication 14 May 2024; date of current version 3 July 2025. This work was supported in part by the National Key R&D Program of China under Grant 2022YFE0137800, in part by the Key "Pioneer" R&D Projects of Zhejiang Province under Grant 2023C01120, in part by NSFC under Grant U22A2032, and in part by the Collaborative Innovation Center of Artificial Intelligence by MOE and Zhejiang Provincial Government (ZJU). The work was also supported by HK RGC GRF under Grant 16210722. This work was done when L. Ying and S. Dou were interns at MSRA. Recommended for acceptance by X. Yuan. *(Corresponding author: Yun Wang.)*

Lu Ying is with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou, Zhejiang 310027, China, and also with Microsoft Research Asia (MSRA), Beijing 100080, China (e-mail: yingluu@zju.edu.cn).

Yun Wang, Haidong Zhang, and Xinyang Jiang are with Microsoft Research Asia (MSRA), Beijing 100080, China (e-mail: wangyun@microsoft.com; haizhang@microsoft.com; xinyangjiang@microsoft.com).

Haotian Li and Huamin Qu are with the The Hong Kong University of Science and Technology, Hong Kong (e-mail: haotian.li@connect.ust.hk; huamin@cse.ust.hk).

Shuguang Dou is with the Department of Computer Science and Technology, Tongji University, Shanghai 200070, China, and also with Microsoft Research Asia (MSRA), Beijing 100080, China (e-mail: 2010504@tongji.edu.cn).

Yingcai Wu is with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou, Zhejiang 310027, China (e-mail: ycwu@zju.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TVCG.2024.3397004, provided by the authors.

Digital Object Identifier 10.1109/TVCG.2024.3397004

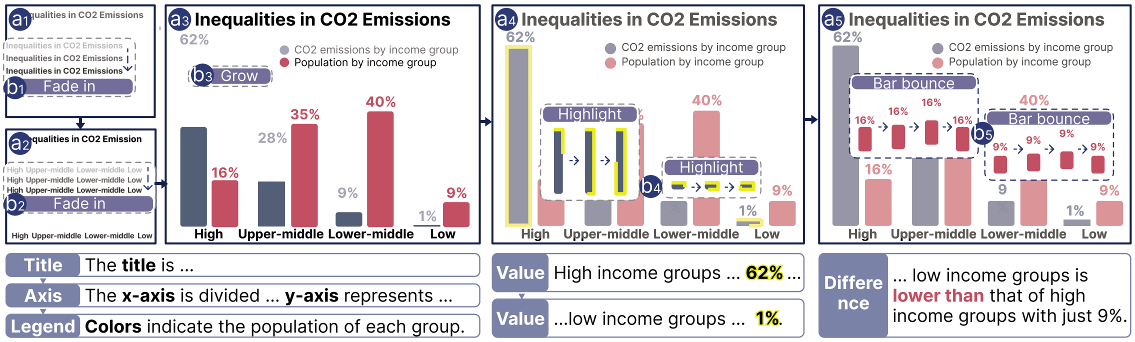


Fig. 1. An example of a Live Chart. (a1-a5) The sequential process of the Live Chart. (b1-b5) The animations in the chart. The texts below are the audio narration for the corresponding frames, with the first tag indicating the chart component and the insight type.

narration. Given that data insights are frequently used to create narratives [15], we expect our narration to contain a compelling data story showcased together with the Live Chart. We employ large language models (LLMs) to uncover intriguing insights within the data and then generate the corresponding narration. Ultimately, we integrate the narration with the static chart to create a Live Chart by animating relevant visual elements based on the extracted insights and corresponding narration.

We evaluate the effectiveness of our approach in different aspects. We first present a variety of real-world use cases that highlight the practical applications of our methods. Following that, we showcase the performance of our chart understanding model, which serves as the foundation for our automated technique. Furthermore, we conduct a crowd-sourced user study to investigate our generated Live Charts and conduct expert interviews to further evaluate our automated approach. Finally, we discuss the insights gained from our research and propose potential avenues for future exploration.

Our primary contribution is an automatic pipeline for reviving static charts into Live Charts. The approach consists of 1) a method including a dual-stream GNN that recovers data in SVG charts and 2) an automatic enhancement from recovered data by incorporating narrations and animations, resulting in the generation of Live Charts. To validate the efficacy of our approach, we conducted comprehensive evaluations. The results provide evidence supporting the effectiveness of our chart revival approach and the resulting Live Chart format.

II. Related Work

In this section, we review research on visualization understanding, visualization enhancement and data video generation.

A. *Visualization Understanding*

The proliferation of visualizations gives rise to research efforts to understand existing images automatically [16]. Given visualization images, researchers seek to understand their content (e.g., chart type, data, and visual encodings). Based on the image input, prior works can be mainly categorized into two classes: raster image and vector image.

Researchers have explored the interpretation of raster images from different perspectives [17], [18]. For instance, Savva et al. [19] proposed ReVision, which identified chart types and extracted data from bitmap charts. Later Davila et al. [16] conducted a survey that covered the topic of automated data extraction from chart images. Despite the support of extracting various information from chart images, raster images are hard to reuse or add extra graphic elements onto the image [20]. Harper and Agrawala [21], [22] focused on vector images and used D3 chart [23] as the input to extract the complete chart structure. With the embedded information, the charts can be utilized in multiple scenes, such as restyle [21], template extracting [22], and style searching [24]. Kim et al. [25] focused on Vega-Lite charts [26] and extracted data and visual encodings in two stages. Recently, Masson et al. [20] proposed an interactive method to extract data from vector charts.

Our approach also takes vector charts as the input since we need to motion the image. We use charts in SVG format with structural information [27]. Moreover, the structures of charts generated by different tools are distinct (e.g., D3, Vega-lite, and Adobe Illustrator). Compared with the methods that targeted a specific SVG type, our model's structure is not influenced by the format of SVG files. Specially, we analyze the structure information for online SVGs automatically by GNN techniques, including the data and visual encodings.

B. *Visualization Enhancement*

Visualization enhancement has attracted researchers' interest in recent years with the evolving techniques of visual understanding [28], which may assist users in understanding the visualization quickly and effectively. The methods can be mainly categorized into three categories, question-answering, annotation, and caption.

Question answering refers to generating answers given a question. Some studies include DQVA [29], FigureQA [11], PlotQA [30] contributed datasets and models for different plots. Further, researchers explored new methods (e.g., LEAF-QA [31], FigureNet [32]) to achieve higher accuracy in question-answering tasks. Compared with treating the input chart image using computer vision techniques, Kim et al. [25] focused on Vega-lite charts and extracted the chart structure to answer natural language questions.

Annotation, as the extra graphical or textual elements [33], plays a vital role in conveying information in data-driven storytelling [6] and attracting users' attention to specific parts of the visualization [34]. Kong and Agrawala [35] presented an automatic system to aid chart reading by generating desired overlays. Bryan et al. [36] proposed Temporal Summary Images to assist visualization analysis via interactive annotation by inputting images and raw data. Lai et al. [12] adopted textual descriptions and images as the input. They introduced an automatic pipeline for annotating visualization. Unlike the raster images in the above work, Lu et al. [37] focused on web-based visualizations and developed a system to augment them via a palette of interactions. Ren et al. [6] characterized a design space of annotation and implemented ChartAccent, which enables users to easily enhance SVG-based charts by a series of annotation interactions.

The caption provides a summary for a visualization. Mittal et al. [38] presented a system, which uses a text planner to select the caption content and structure. Recently, researchers utilized template-based descriptions to depict charts in certain genres of narrative visualization [39], such as slideshows [40]. With the development of computer vision and natural language processing techniques, researchers contributed plenty of models to generate accurate descriptions given rasterized charts [41], [42]. Liu et al. [43] targeted vector images and proposed an approach including several models to generate captions automatically.

Our work aligns with the direction to augment the chart but goes beyond to revive the charts into Live Charts.

C. Data Video Generation

Plenty of research is dedicated to enhancing data visualization through creating data videos [44]. Many studies focus on integrating animations and developing systems to generate animations from data [45], [46], [47]. For example, DataClips [45] allows non-experts to generate data videos by inputting data. Through a series of interactions, users can select and integrate data-driven clips from the given library and form a video. DataParticles [48] allows creators to author animated unit visualization in data stories. AutoClips [46] moves forward to save human effort by automatically crafting data videos from a sequence of data facts. However, the above work cannot revive static charts with image input. There are alternative approaches that begin with specific image inputs. For instance, Data Animator [49] chooses the Data Illustrator [50] file format as the input, which includes detailed data and visual encoding information and enables authors to generate animated data graphics and transitions without programming. Based on Canis [51], Ge et al. [52] designed CAST, an interactive authoring tool to create data videos by inputting data-enriched SVGs. However, these methods have specific input requirements and cannot process SVGs without data.

Though these tools assist users in generating animations, much effort is still required to transform static charts into Live Charts. Focused on infographics, Infomotion [10] allows users to automatically generate animated presentations from static infographics. However, it does not assist basic charts. Moreover, current studies tend to neglect the importance of audio narration, which is a crucial aspect of creating a comprehensive data video. As a result, users must still use additional tools to record audio narration and synchronize it with animation. In our research, we propose methods for automatically deriving Live Charts from static ones and consider both the animation and audio narration simultaneously in the process.

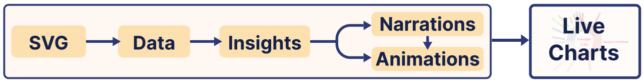


Fig. 2. Our automatic approach for generating live charts.

III. Live Charts

Since making Live Charts from static vector-based charts requires considerable effort, we aim to revive them into Live Charts automatically.

To enhance the chart expression, previous studies have emphasized the significance of annotations or textual cues to enhance user comprehension of charts [6]. In contrast to traditional text cues near charts that divert visual attention, we utilize voiceover narration–freeing up visual focus while enhancing chart comprehension. Moreover, prior research [9] highlights the effectiveness of breaking down complex information into smaller, more digestible chunks when presenting charts. In this regard, we choose to integrate animations into Live Charts, which can assist users in visually processing intricate data and comprehending complex chart representations. Furthermore, a multi-sensory experience by synchronizing narrations and animations within the Live Charts can complement each other by providing supplementary details, or indirectly link to each other by presenting data from different perspectives [53]. Therefore, we propose an approach (Fig. 2) that involves three steps:

1) Using GNN-based techniques, we identify the data together with the visual encodings from the input SVG.
2) We derive the audio narration with insights and its corresponding data with LLMs.
3) For some expressions in narration, we design the corresponding animations and synchronize them to the audio.

We then describe each step in detail accompanied by an example depicted in Fig. 3(a), which we refer to as the “airport chart.”

A. Chart Understanding

In this section, we present an approach for automatically understanding charts and recovering data together with‘ visual encodings.

Compared to common existing methods that rely on pixel-based charts, our proposed approach directly handles the raw SVG files to revive charts into Live Charts. Pixel-based methods often suffer from high computational overheads and limitations in modifying individual elements due to their rasterized nature [54], [55]. In contrast, SVG files, being vector-based, offer efficient rendering regardless of resolution and allow for targeted modifications through the use of element properties. The structured information between elements in vector graphics allows us to model SVG elements, such as lines, paths, and circles, as a multi-graph, which effectively captures the essential

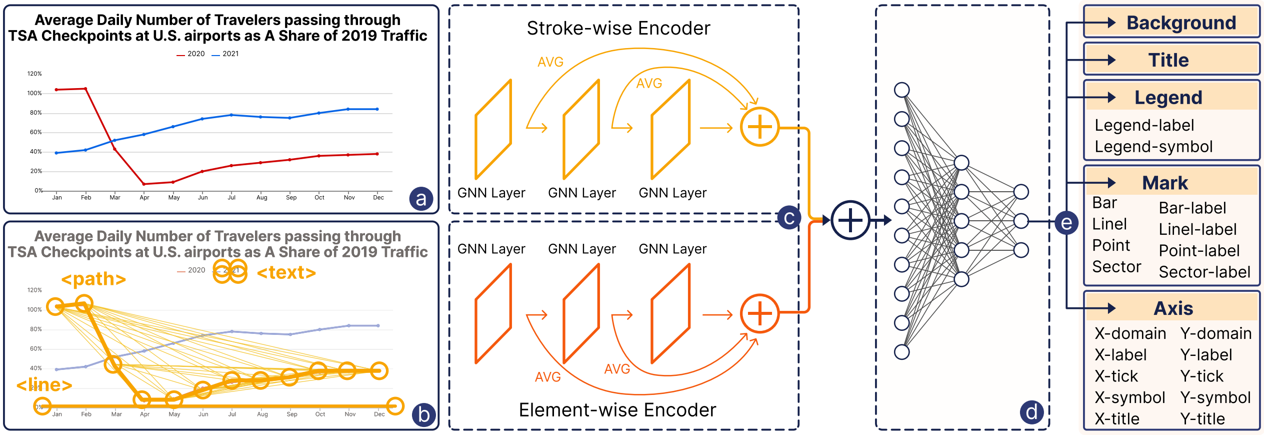


Fig. 3. A GNN-based approach to classify chart elements for a chart. (a) Original SVG-based chart; (b) Conversion from SVG to graph; (c) Two different vector encoders to extract features; and (d) Classification of each element by a multi-layer perceptron. (e) Categories for elements in the chart. Each box corresponds to a primary category, followed by a set of sub-categories.

structural and spatial information. In light of the proficiency of GNNs in handling structured graph-based data and their success in previous research [27], we leverage GNNs for recognizing and analyzing elements within SVG files.

Especially, our chart understanding method involves a two-stage process: recognizing chart elements (e.g., marks, legends) and recovering visual encodings. First, we apply a GNN-based method [54] to detect chart elements and classify them into predefined categories. Second, we establish a set of rules to extract data and recover visual encodings based on the results from the classification.

1) Chart Element Recognition: Our goal is to identify various chart elements within a given chart. To utilize the structural information within SVGs, we first convert the chart to graphs and apply a GNN-based method to extract features with two vector encoders. Then, we predict each element's category based on the extracted features.

Convert Chart to Graph: We begin by constructing nodes and then connecting them with edges. SVG-based charts are represented as a set of SVG elements, including path, text, line, and other basic elements. Two types of SVG elements (graphic and textual) are handled separately, as they possess distinct attributes. For graphic elements such as paths and lines, we utilize each element's start and end points as two nodes of the corresponding graph. For text elements, we connect the vertices of the bounding box to form a 4-node graph. Yellow circles in Fig. 3(b) provide examples of the nodes. Next, a feature vector consisting of five dimensions was created to represent each node, including the element type (e_t), which is converted into a one-hot vector, the spatial coordinates of the points (pos), the fill color (e_f), the stroke color (e_{sc}), and the stroke width (e_{sw}). Thus, the node feature V_f is:

$$V_f = \{e_t, pos, e_f, e_{sc}, e_{sw}\}. \tag{1}$$

After constructing nodes, we proceed to connect them using two types of edges. The first type is stroke-wise edges (E_s), which connect nodes that are directly connected in the SVG, such as adjacent points in a line chart. These edges are depicted by thick yellow lines in Fig. 3(b).

$$E_s = \{(v_i, v_j) | v_i, v_j \in \mathbb{C}\}, \tag{2}$$

where $\mathbb{C}$ is a set of tuples containing the start point v_i and end point v_j.

The second type, element-wise edges (E_e), connect nodes that have the same element type e_t. These connections can be observed as thin yellow lines in Fig. 3(b):

$$E_e = \{(v_i, v_j) | v_i, v_j \in e_i\}, i \in N. \tag{3}$$

After that, we obtain the graph $G(V, E_s, E_e)$ with several nodes and the connected edges, where V denotes the node, E_s denotes the stroke-wise edge, and E_e denotes the element-wise edge.

Feature Extraction With Two Vector Encoders: After converting the chart into a graph, we extract features from it using two vector encoders, each tailored for the corresponding category of edges. For stroke-wise edges, we utilize a stroke-wise encoder, using a GNN layer that incorporates the functions of two linear transformations. For element-wise edges, given the larger number of edges, we employ a GNN layer with one linear transformation. The specific architecture of the two encoders is depicted in Fig. 3(c), where the features derived from each layer of the encoders are aggregated and utilized for downstream tasks.

$$F_{fuse} = concat(f_s^1, f_s^2, \ldots, f_s^l, f_e^1, f_e^2, \ldots, f_e^l), \tag{4}$$

where F_{fuse} is the fused features of a specific element.

Classification Task: For each SVG element, we obtain the graph representation by aggregating the features extracted by two vector encoders. To comprehend the categories that are depicted by each element, we utilize a multi-layer perceptron to classify the graph representation that corresponds to each element. We define five primary categories: background, title, legend, mark, and axis. The corresponding sub-categories can be found in Fig. 3(e). The classification task is performed across these class labels, with cross-entropy losses serving as the training objective for the entire model.

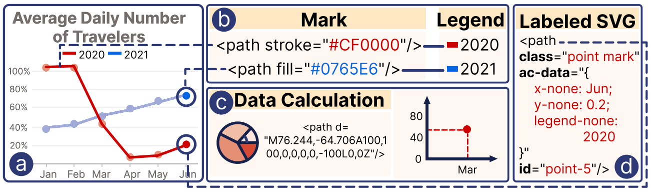


Fig. 4. The process of recovering visual encodings. (a) The example chart. Dotted dark blue lines connect chart elements with their SVG expressions. (b) Matching the mark with the legend. (c) Data calculation for different charts. (d) One path in the labeled SVG.

Dataset: To train the above model, we require an SVG chart dataset that encompasses different types of charts. Additionally, the ground truth information for each element is essential for indicating its corresponding category in the chart. We decided to use Vega-Lite [26], Plotly [1] and D3 [23] for generating charts instead of collecting due to the substantial amount of data required. The high-level grammar enables the creation of charts in various types and styles while striking a balance between customization and complexity [26]. We utilized the JavaScript library "Faker"[2] to populate the charts with synthetic data. In this paper, we consider three basic chart types: bar chart, line chart, and pie (donut) chart. We conducted a comprehensive analysis of different chart components, such as background, axis, and points, based on previous collections [6]. We aim to identify the diverse styles associated with each component. Following that, we made a comprehensive list of styles, which can be used for generating charts. Consequently, for Vega-lite, Plotly, and D3, we generated 3,000 charts for each chart type separately, resulting in a total of 9,000 charts for each dataset .[3] The performance of our model on this dataset is evaluated in Section IV-B.

2) Data and Visual Encoding Recovering: We aim to recover the data and visual encodings for the above categorized elements. This is achieved by analyzing the relationships between the classified elements in the SVG. We then add the data and visual encodings into the original SVG and produce a labeled SVG to facilitate the next two steps for generating the Live Chart. The process contains three steps:

Legend-Mark Matching: With the aid of the classification results, we can obtain the bounding box of the legend labels and legend symbols through the attribute d in the path element. For charts that include legends, we determine the corresponding label for each legend symbol by calculating the euclidean distance between the top-left corner of the bounding box of each symbol and label and identifying the nearest label. Next, we match each mark with the corresponding legend by comparing the stroke and fill color of the mark and legend symbols (Fig. 4(b)).

Data Calculation: For charts that contain textual information, such as bar labels appearing alongside the bars, we match each mark with the closest label to retrieve the corresponding y-value. For charts that lack textual information, we handle the Polar (pie charts) and Cartesian (line and bar charts) coordinate systems separately, as shown in Fig. 4(c). For the Polar coordinate system, we parse the d attribute for the path element and get the arc length of all sectors. We then calculate the percentage of data represented by each sector. For the Cartesian coordinate system, the marks are first matched with the closest "x-label" in horizontal distance to retrieve the corresponding x-value. Then, the y-value is calculated using an interpolation method with all values of the "y-label" and the vertical distance between the marks and the closest"y-label".

Output Labeled SVGL: After we calculate the data within the chart, we add it to the original SVG. The output labeled SVG C_{lsvg} contains the same structure as the original SVG, but with additional attributes that represent the extra information extracted from the chart. Fig. 4(d) shows an example ¡path¿ in the labeled SVG. We add three attributes:

- *Class:* We assign two class names, including the primary category and the sub-category type (Fig. 3(e)), to the element.
- *ID:* We define the ID as the concatenation of the sub-category type and an ordered number (e.g., *line-0*). This attribute is used for identifying specific elements within the SVG in the following workflow.
- *Ac-data:* We store the extracted data in this attribute, using two data dimensions for charts without legends and three dimensions for charts with legends. We assign the content of x-title, y-title, and legend-title as the field names for these dimensions. In cases where these fields are empty, such as the x, y, legend fields in the line chart depicted in Fig. 4(a), we label them as "None" (Fig. 4(d)).

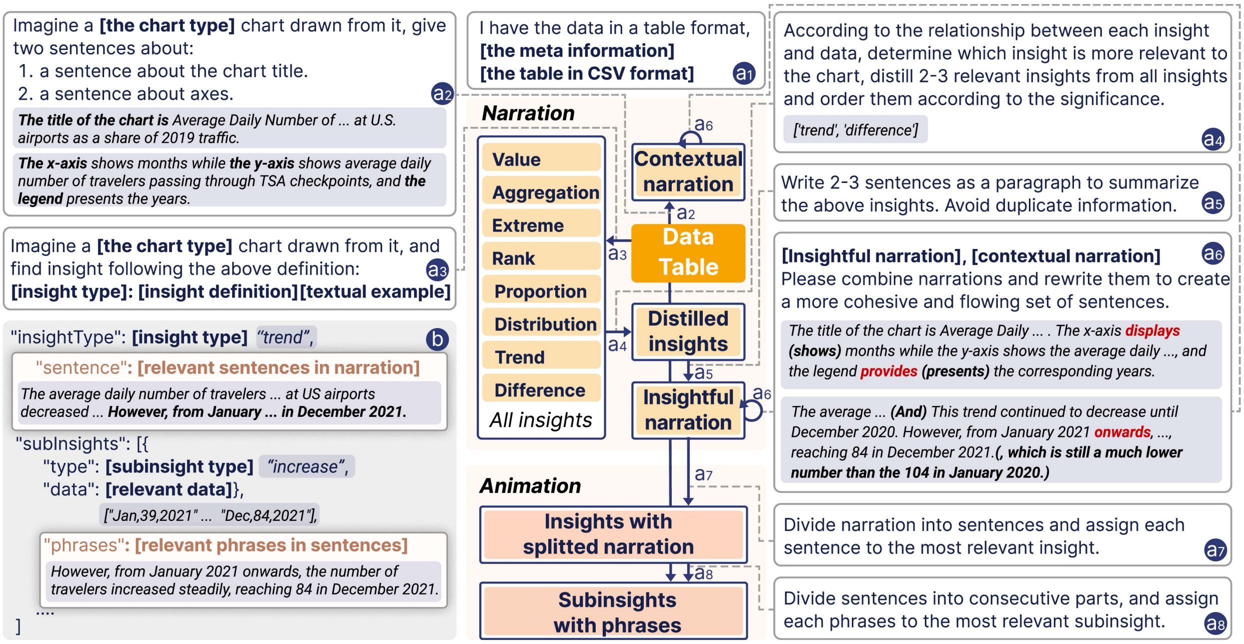
B. Narration

In this section, we introduce methods to generate audio narration. High-quality audio narrations can add informative context and direct users' attention [56]. From the previous analysis [53], narration often involves data contexts and insights. We accept a similar structure: contextual information following interesting insights.

Researchers have adopted various methods to generate insights [57], [58], [59]. Inspired by the remarkable proficiency of large language models (LLMs) in handling open-ended queries, we select GPT-3 to generate narrations. When presented with a text $prompt$, the GPT-3 model generates a string-format answer. Adhering to the fundamental principles [4] of "show and tell", "provide quality data", and "check settings", we devise distinct prompts for the following various tasks.

Since GPT-3 cannot process images directly, we use the chart specification as input. We create a data table by extracting the "ac-data" attribute from the labeled SVG. In addition, we gather meta information such as the chart's title, x-axis, and y-axis labels. This data table and meta-information serve as the foundational components of each prompt (Fig. 5(a1)). Considering the token limitation of prompts (the maximum number of words or tokens that can be used as input to the model at once),

[1] https://plotly.com/
[2] https://fakerjs.dev/
[3] https://github.com/YiYinYinguu/SVG-Chart-Dataset
[4] https://beta.openai.com/docs/guides/completion/prompt-design



Imagine a **[the chart type]** chart drawn from it, give two sentences about:
1. a sentence about the chart title.
2. a sentence about axes.

The title of the chart is *Average Daily Number of ... at U.S. airports as a share of 2019 traffic.*

The x-axis *shows months while* ***the y-axis*** *shows average daily number of travelers passing through TSA checkpoints, and* ***the legend*** *presents the years.*

Imagine a **[the chart type]** chart drawn from it, and find insight following the above definition: **[insight type]: [insight definition][textual example]**

"insightType": **[insight type]** *"trend",*
"sentence": **[relevant sentences in narration]**
The average daily number of travelers ... at US airports decreased ... ***However, from January ... in December 2021.***
"subInsights": [{
"type": **[subinsight type]** *"increase",*
"data": **[relevant data]**},
["Jan,39,2021" ... "Dec,84,2021"],
"phrases": **[relevant phrases in sentences]**
However, from January 2021 onwards, the number of travelers increased steadily, reaching 84 in December 2021.
....
]

I have the data in a table format, **[the meta information]** **[the table in CSV format]**

According to the relationship between each insight and data, determine which insight is more relevant to the chart, distill 2-3 relevant insights from all insights and order them according to the significance.

['trend', 'difference']

Write 2-3 sentences as a paragraph to summarize the above insights. Avoid duplicate information.

[Insightful narration], [contextual narration]
Please combine narrations and rewrite them to create a more cohesive and flowing set of sentences.

The title of the chart is Average Daily ... . The x-axis ***displays (shows)*** *months while the y-axis shows the average daily ..., and the legend* ***provides (presents)*** *the corresponding years.*

The average ... ***(And)*** *This trend continued to decrease until December 2020. However, from January 2021* ***onwards***, *..., reaching 84 in December 2021.****(, which is still a much lower number than the 104 in January 2020.)***

Divide narration into sentences and assign each sentence to the most relevant insight.

Divide sentences into consecutive parts, and assign each phrases to the most relevant subinsight.

Fig. 5. Creating narration and animations involves multiple sub-tasks for GPT-3, depicted by the inner blue boxes and arrows. The starting position of the arrow indicates the information contained within the prompt (with "Data Table" being a necessary element for each prompt). Each blue box represents the output generated by GPT-3. (a1)-(a8) are part of prompt components. We use dotted gray lines and numbers to illustrate the prompt usage during the process. (b) The insight structure in JSON format. Sentences and phrases will be added in animation generation. Italic text with a blue background represents an example of the prompt output, using the "airport chart" in Fig. 3(a), to demonstrate how the process works.

we represent the table in a CSV format. Moreover, we follow established design guidelines [60] and develop distinct prompts tailored to each specific task.

1) Narration With Contextual Information: Our narration starts with a sentence that describes the chart. We follow the "Level 1 semantic content" framework proposed by Lundgard [61], which specifies that the description should encompass the chart type, title, legend, encoding channels, and axis information. Thus, we design the prompt as illustrated in Fig. 5(a2), incorporating the basic chart information from Fig. 5(a1) to fulfill our requirements. The resulting output of this step from GPT-3 will consist of two sentences: the first about the chart type and title and the second addressing the legend, encoding channels, and axis information. The example sentences are depicted in Fig. 5(a2).

2) Narration With Insights: Engaging insights is the foundation of a high-quality narration [15]. Following the definitions of facts provided in a prior study [62], we have eight facts as prospective insights for three types of charts: value, proportion, difference, trend, rank, aggregation, extreme, and outlier. Within each insight type, there can be various sub-types. For example, the proportion insight may encompass majority and minority data. Consequently, we expand the sub-insights, as detailed in Table I, based on the definitions of these eight insights.

In the beginning, we expected that GPT-3 would produce several relevant insights from the data table. However, this complex task requires a significant amount of input text comprising multiple sentences, which may exceed the maximum allowable number of tokens. Consequently, we subdivide the task into three sub-tasks: generating all insights, distilling insights, and creating narration.

TABLE I
The Table Presents the Insight Type and Corresponding Sub-Insight Types

Insight Type	Sub-insight Type
Proportion	Majority, minority.
Extreme	Maximum, minimum.
Distribution	Normal, uniform, none.
Aggregation	Average, sum, count.
Trend	Fluctuate, increase, decrease, fluctuate increase, fluctuate decrease.

We first generate all insights with individual prompts for each insight. These prompts consist of insight definitions (Fig. 5(a3)) referenced from previous research [46], [62] and basic chart information (Fig. 5(a1)). To ensure efficient data analysis, we require GPT-3 to generate JSON output (Fig. 5(b)) [63]. Sometimes, extra explanatory text is added to the output, so we utilize regular expressions to preserve the intended format. Examples can be found in the supplementary. In this step, we ask GPT-3 to simultaneously identify relevant data for each sub-insight. From our experiment and recent researches [64], GPT-3's proficiency in handling mathematical computations, particularly those involving complex calculations (*extreme* and *rank* insights), is limited. To address this, we introduce a validation procedure to verify output values. In cases where inaccuracies are identified, heuristic approaches are applied to generate insights, as demonstrated in the supplementary material.

To craft a compelling data narrative, we focus on the salient insights generated by GPT-3. Thus, we aim to distill insights. With all insights and the basic chart information (Fig. 5(a1)), we task GPT-3 with comparing the relationship between each insight and the visual representation of the data (Fig. 5(a4)). After processing the input data, GPT-3 will generate an output that lists several relevant insight types. Taking the example of the "airport chart", the result is *"trend"* and *"difference"*.

We then create the narration. A captivating narrative should interconnect each sentence and flow cohesively rather than being disjointed and independent. Therefore, we feed all distilled insights into the prompt (Fig. 5(a5)), to elicit a coherent and clear narration from GPT-3.

3) Rephrase the Narration: After acquiring the two segments of narration, we combine them as a new prompt for GPT-3 to refine the final narration (Fig. 5(a6)). We expect the sentences in the two parts to have smooth transitions between them. Keeping the original meaning and essential words, such as the title and axis, GPT-3 will create a refined and cohesive narration by seamlessly connecting the contextual and insightful sentences. Moreover, we require GPT-3 to re-assign the narration as two parts for the next step. As shown in Fig. 5(a6), the red words indicate the adjusted version, and the words in the brackets are omitted. Finally, we derive a new narration for contextual information and a new narration for insights.

We then leverage text-to-speech techniques (Azure TTS API[5]) to convert the narration into an audio narrator.

C. Animation

Previous research has demonstrated that animations synchronized with narration can stimulate both visual and auditory senses, leading to improved data representation [53], [65]. Therefore, we describe how we incorporate animation and synchronize it with narration in this section. We design the animations based on the two-part narration generated in Section III-B. We aim to identify the data related to specific phrases in the narration and add appropriate animations to the corresponding time intervals. Based on prior research [51], we use three key aspects to define an animation:

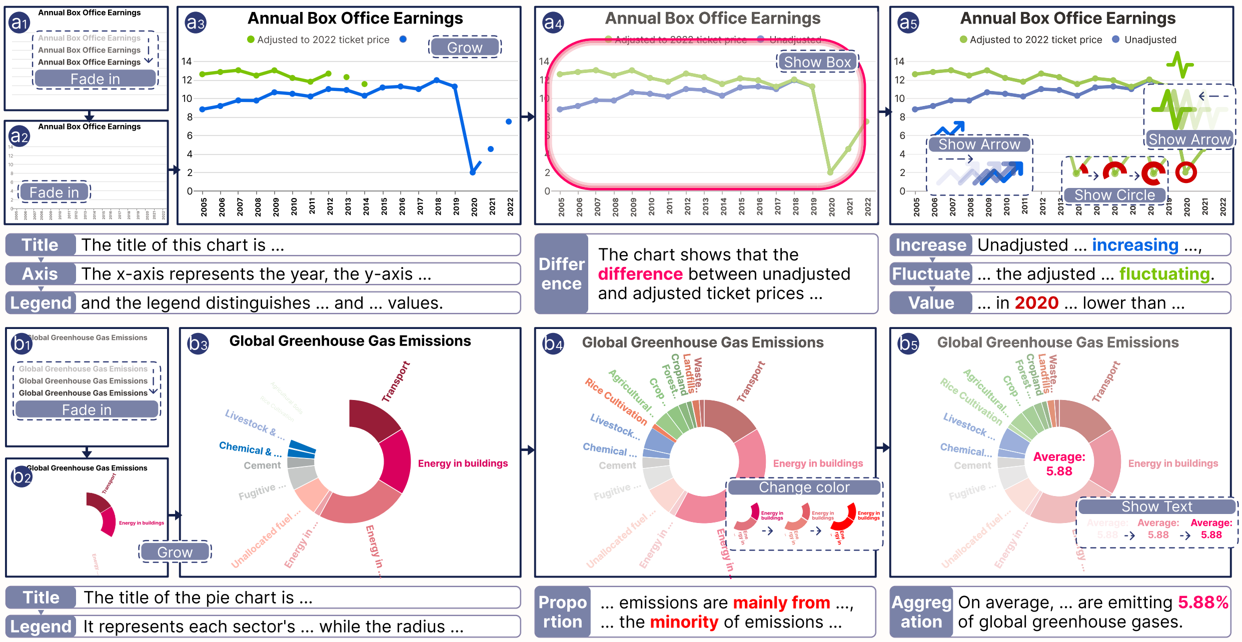
- *Target:* The target elements to be animated. As we have added relevant classes and IDs into the input SVG as the labeled SVG (as discussed in Section III-A), targeting elements is not challenging. Technically, we utilize the W3C Selectors API.
- *Effects:* The animation effects. Similar to [10], we adopt three principal animation types in presentation software (e.g., Keynote and PowerPoint): entrance, emphasis, and exit. Then, we choose multiple prevalent animation styles from past research [10], [51] and devise styles to fulfill our requirements. Part of the animation effects is depicted in Fig. 6 with dotted blue boxes.
- *Interval:* A suitable time interval for elements to enter and exit the screen. Instead of using time seconds to represent the entry and exit times, we adopt a different approach. We use the start and end word indices in the narration as a reference. For instance, the interval $[10, 20]$ denotes the animation starting from the 10th word and ending at the 20th word. Since we use a text-to-speech technique to generate the audio, the API can conveniently generate the timestamp of a specific word in the narration. Thus, with all the words having timestamps, we can convert the word index interval into a normal time interval.

Defining the three variables is the key to completing the task of adding animation. For each animation, we need to identify the relevant phrases first and convert them into a word index interval. Then, we select the corresponding elements as targets and apply an appropriate animation effect to create a polished and visually engaging Live Chart. We process the two-part narration separately.

1) Animation for the Contextual Narration: We first analyze the contextual narration to locate the interval and identify the relevant targets. For the first sentence related to the chart title, we set the target as *"title"* and locate the entire first sentence as its interval. Moving on to the second sentence about encodings, we locate the position of the words "x-axis" and "y-axis" for bar and line charts. The sentence can be divided into three parts, each representing a distinct interval. The first and second parts target *"x-axis"* and *"y-axis"*, respectively, and the last part target *"legend"*. For instance, as shown in Fig. 5(a6), the first part is represented by "displays months while", the second part is represented by "shows the average... and". For pie charts, we set the interval as the whole sentence and the target as *"sector"* and *"legend"*. To display corresponding elements in the front part of live charts, we utilize the entrance animation among the three available animation types. Thus, animation effects like fade-in and wipe are selected for the contextual narration.

2) Animation for the Insights: We create animations that correspond to each insight in order to enhance the storytelling. Having identified the pertinent data for each sub-insight, we utilized the "ac-data" attribute labeled in Section III-A to target the appropriate elements. SVG paths and texts can be the candidates. Concerning the word interval, we continue to rely on GPT-3 to interpret the insightful narration, as it is responsible for generating the narration itself. Our approach is progressive, consisting of two prompts. Given the table, a list of the distilled insights, and insightful narration, the first prompt is employed to assign sentences to each insight (Fig. 5(a7)). In the output, sentences are included in the insight structure (Fig. 5(b)). Furthermore, with the same inputs (table and insightful narration) and the distilled insights accompanied by their corresponding sentences, the second prompt is utilized to allocate phrases within each insight sentence to each sub-insight (Fig. 5(a8)). Phrases are added in the subinsights structure after this step (Fig. 5(b)). Through this process, we identify the appropriate word interval for the animation. Then we proceed to select the animation effects for the above elements. We opt for emphasis animations while choosing animation effects. We use the animation effects "show box", "show text", "change color" and "highlight" for the elements in three chart types. Additionally, we include the "bar bounce" effect for bar charts, as well as "show arrow" and "show circle" for line and bar charts. Sometimes certain animation

[5] https://azure.microsoft.com/services/cognitive-services/text-to-speech



Title The title of this chart is ...

Axis The x-axis represents the year, the y-axis ...

Legend and the legend distinguishes ... and ... values.

Difference The chart shows that the difference between unadjusted and adjusted ticket prices ...

Increase Unadjusted ... increasing ...,

Fluctuate ... the adjusted ... fluctuating.

Value ... in 2020 ... lower than ...

Title The title of the pie chart is ...

Legend It represents each sector's ... while the radius ...

Proportion ... emissions are mainly from ..., ... the minority of emissions ...

Aggregation On average, ... are emitting 5.88% of global greenhouse gases.

Fig. 6. Two use cases. (a1-a5)(b1-b5) The image flow illustrates the keyframes of the Live Chart. Animations are drawn with dotted blue boxes. The following text describes the corresponding audio narration, with the first tag indicating the chart component or the insight type.

effects may not have specific elements to target within a chart. For instance, the “show circle” effect cannot be used for line charts that lack points. In such scenarios, we alter the emphasis effect into a list of entrance-exit animations. To be precise, we create such elements and allow them to enter at the beginning of the animation, followed by their exit at the end of the animation. We select different animation effects for different insights in each chart to ensure the richness of the generated charts. All animation effects are chosen based on the chart type and insight type.

IV. Evaluation

In this section, we first present a set of use cases for reviving real-world charts into Live Charts. We describe our evaluation process, which involved testing the performance of the understanding model and conducting a user study and expert interview to evaluate Live Charts. Our evaluation results further confirm the effectiveness of our workflow and demonstrate the potential of the Live Chart format.

A. Use Cases

To demonstrate the capabilities of our workflow in Section III, we present a set of Live Charts, including bar, line, and pie (donut) charts. We collected chart SVGs from the Internet and ensured that they covered different topics and styles. Figs. 1 and 6 show part of the examples in our gallery.[6] Each chart is presented in a step-by-step image sequence from index 1 to 5, beginning with contextual information (title, axis, and legend) and followed by various insights. We use an animation box for each insight depicted in the chart, using a dotted stroke (as shown in Fig. 6), to indicate both the type of animation effect and the process by which the corresponding element is altered. Below the image sequence, we provide the corresponding narration with a tag to indicate contextual or insightful information, followed by a specific description.

B. Performance of Chart Element Recognition

We conducted a series of quantitative evaluations to assess our understanding method in comparison with baselines.

We selected pixel-based methods as our baselines, utilizing mmyolo [66] for implementation while keeping all parameter settings remaining the same. Our model is implemented using PyTorch Geometric [67] built upon PyTorch [68]. We construct a two-layer GNN for the stroke and element-wise encoder. We evaluated different machine-learning model parameters and settled on the following sets. The hidden node representation dimension is set to 64, as deeper GNNs tend to suffer from over-smoothing. To avoid over-smoothing, mean aggregation is only conducted on the final layer of the GNN, and each node requires only one transformation and one mean-pooling operation. Moreover, a three-layer MLP is employed as the classifier, with the output dimensions of the middle layers set to 512 and 256. The Adam optimizer is utilized with a learning rate of 0.001 and a batch size of 128. The training procedure is performed from scratch for 200 epochs on an Nvidia V100 GPU card.

Evaluation Metric: We adopt the commonly used metrics of AP50, AP75, and mAP. Specifically, AP* denotes the average precision at the intersection over the union (IOU) threshold of 0.5 or 0.75 for object detection tasks. Moreover, we calculate

[6] The whole examples can be found in the supplementary materials.

TABLE II
THE TABLE PRESENTS THE EFFECTIVENESS OF PIXEL-BASED METHODS AND OUR DUAL-STREAM GNN SPECIFICALLY DESIGNED FOR VECTOR GRAPHICS RECOGNITION ON VEGA-LITE, PLOTLY, AND D3 DATASETS

	Vega-lite			Plotly			D3		
Methods	AP_{50}(%)	AP_{75}(%)	mAP (%)	AP_{50}(%)	AP_{75}(%)	mAP (%)	AP_{50}(%)	AP_{75}(%)	mAP (%)
YoloV8-m	77.30	68.90	64.60	77.60	70.60	**66.00**	64.30	50.50	49.30
Ours (E_e)	71.75	68.65	63.14	69.25	64.90	54.93	61.43	59.77	51.98
Ours (E_s)	84.81	80.75	74.28	78.04	74.06	62.87	62.47	60.27	52.66
Ours ($E_s + E_e$)	**85.71**	**81.85**	**75.21**	**80.04**	**76.57**	64.55	**64.77**	**61.23**	**53.40**

E_e denotes only using the element-wise encoder, E_s denotes only using the stroke-wise encoder, and $E_s + E_e$ denotes using the two encoder.
The best results when performing each method are in bold.

the mAP as the mean of the average precision over the IOU thresholds ranging from 0.50 to 0.95.

Results: We use the dataset described in Section III-A1. For pixel-based methods, we compare our methods with popular one-stage object detection method YoloV8 [69] based on mmyolo. For YoloV8, the -m variant is scaled YoloV8 with more parameters and better performance. As shown in Table II, we evaluate the effectiveness of two different vector encoders for chart element recognition. When solely relying on the stroke-wise encoder, our model demonstrates commendable performance. Through our experimentation, we observed that using both stroke-wise and element-wise encoders in conjunction, via the GNN, results in optimal classification performance.

C. User Study

We further assessed the effectiveness of Live Charts generated by our approach through a user study with 90 participants. We conducted our study on Prolific ,[7] a popular platform for research participation. This study aims to assess the following aspects:

- Can Live Charts assist users in understanding charts?
- In what ways do animation and narration in Live Charts enhance the chart representation?

We describe the procedure and results in this section.

1) Participants: We used the Qualtrics survey platform to set up a crowd-sourcing experiment. 90 participants (29 females) were recruited from Prolific in our study. The majority of participants, comprising 42%, fell into the age range of 25 to 34 years, followed by 27% in the age range of 35 to 44 years. As the task required writing, participants' proficiency in writing and English vocabulary could influence the outcomes. Thus, we limited the study to only English-speaking users who had previously received at least a 95% approval rate on their results. We compensated each participant with $4.9 for completing the 40-minute task.

2) Procedure: The study consisted of five parts. The first part involved explaining the study procedure and obtaining consent and demographic information from the participants. To ensure that our participants had a basic understanding of insights and visualization, we presented eight insight types that we used in our generation process, along with their definitions and examples. Subsequently, we asked three questions, in which participants needed to observe a chart and answer a question related to three randomly selected insights out of a total of eight. For example, one of the questions could be, "What trend best describes the line?" Only participants who answered these questions correctly were eligible to participate in the study.

To investigate the role of animations and narrations, we carried out a between-subject study. Participants were randomly divided into three groups and presented with varying chart formats: static, animated, and Live Charts. Each participant was only shown one type of chart format. For the third part, we demonstrated a bar, pie, and line chart, providing sample insights for each. For the fourth part, participants were tasked to provide insights for charts. The chart format used was the same as in the previous step. The six charts are presented randomly. Participants were asked to fully explore the chart insights and then write down insights. Moreover, participants with animated charts and Live Charts were required to remain on the page longer than the video length to ensure they viewed it at least once. Additionally, for participants with Live Charts, we included task questions at random intervals to test the audio functionality, to verify that they can hear the accompanying narration. we recorded the insights written by users and further calculated the sentence number of insights.

After completing the writing tasks, participants were asked to score the chart. They were required to rate the chart in five dimensions: insightful, memorable, focused attention, understandability, and enjoyment. We used a seven-point Likert scale, with options ranging from strongly disagree (1) to strongly agree (7). Finally, participants were asked to provide a brief explanation for their rating.

3) Stimuli Preparation: We collected SVG charts from use cases (Section IV-A), covering a wide range of topics. Considering the potential for user fatigue during prolonged experimentation, we prepared a set of 9 charts for each user, with 3 charts for each chart type (bar, pie, and line). Among these, one chart of each type (3 in total) was used for demonstrating examples, and the remaining 6 charts (2 for each type) were used in the formal study. For each chart in the study, we prepared three different formats: a static chart, an animated chart, and a Live Chart. The original SVG was used as the static version, while the Live Charts were generated using the techniques discussed in Section III. For the animated charts, we used the same animations as in the Live Charts. We further shortened the intervals between each animation to one second because sorely removing the narrations from the live version may result in long intervals between animations, which could have bored users.

[7] https://www.prolific.co/

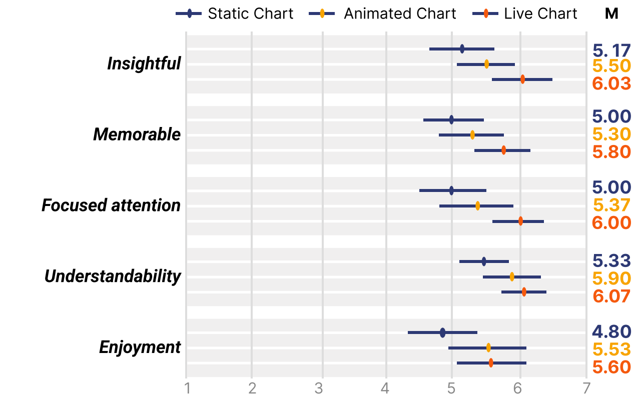


Fig. 7. The statistical analysis results for the overall evaluation include the corresponding Likert ratings. The means are depicted using shapes, and their confidence intervals are shown with error bars. On the right side are the specific numerical mean values.

TABLE III
Overall Statistical Evaluation Result for Each Dimension, Including the Chi-Square Values, P-Values, Degrees of Freedom, and Effect Size

Dimensions	Chart Format Pair	p-value	effect size (r)
Insightful $\chi^2(2) = 8.05$, p=0.017	Static - Animated Static - Live Animated - Live	0.182 **0.008** 0.069	0.17 **0.34** –0.23
Memorable $\chi^2(2) = 6.17$, p=0.045	Static - Animated Static - Live Animated - Live	0.284 **0.018** 0.120	0.13 **0.31** –0.20
Focused attention $\chi^2(2) = 8.85$, p=0.012	Static - Animated Static - Live Animated - Live	0.206 **0.003** 0.087	0.16 **0.38** –0.22
Understandability $\chi^2(2) = 6.36$, p=0.042	Static - Animated Static - Live Animated - Live	0.060 **0.017** 0.733	0.24 **0.31** –0.04
Enjoyment $\chi^2(2) = 6.18$, p=0.046	Static - Animated Static - Live Animated - Live	0.086 **0.013** 0.758	0.22 **0.32** –0.04

The best results for different conditions are in bold.

4) Results: This study gathered the Likert ratings from five perspectives. As all data are not normally distributed, we use the Kruskal-Wallis test for each factor. Additionally, we utilized the Mann-Whitney tests with Bonferroni correction to compare pairs within each factor and accounted for multiple comparisons. The sample means, and their corresponding confidence intervals are depicted in Fig. 7 and Table III.

The Likert ratings, measured on a scale from 1 to 7, showed significant differences across all dimensions ($p < 0.05$). Post-hoc testing revealed significantly higher scores for all dimensions in the live format compared to the static charts ($p < 0.05$, $|r| > 0.3$). Furthermore, we found that Live Charts scored marginally significantly higher than animated charts in the *Insightful* and *Focused Attention* dimensions. Animated charts scored marginally significantly higher than static charts in the *Understandability* and *Enjoyment* dimensions. The results indicate that Live Charts offer a compelling and memorable approach and effectively direct users' attention toward important information.

5) Feedbacks: Based on our observations, most participants in Live Charts provided positive feedback about Live Charts.

Both participants in animated charts and Live Charts acknowledged the effectiveness of animation in conveying data effectively. First, participants using the Live Charts format rated the charts significantly higher in terms of *Understandability* than participants in the static charts format. Animated chart users thought that animations allow for breaking down complex information into smaller, more manageable parts, which in return increases comprehension. Live Chart users also mentioned that *"(Live Charts) provide a way to communicate quantitative information in a way that is easier to understand for many people."* In contrast, some participants in static charts found it challenging to absorb all the information presented in a static chart. For example, one reported that *"A chart might be too complicated that I can't recognize all the patterns."* Moreover, animations provide extra value in the form of accompanying text by extracting information from the static chart using a model. This feature was frequently mentioned as a drawback of static charts, with users reporting difficulty in predicting precise numbers. Second, animated charts and Live Charts users highlighted that animation helps emphasize specific chart elements, transforming data into engaging and memorable stories. For example, one participant in Live Charts mentioned that *"The animation is also great for highlighting trends and turning the data into a memorable story,"* while another participant in static charts wrote that *"(Static charts) do not allow highlighting of specific sections."* Furthermore, users with animated and Live formats agreed that the formats were interesting, while participants with static charts found it dull to look at for a long period. This pattern was also supported in the Likert score for the *Enjoyment* dimension.

Narration emerged as another crucial component, as users noted its ability to complement the animation by directing auditory attention to specific elements. This pattern can be reflected in the higher *Focused Attention* score in Live Charts. Moreover, with narration, Live Charts can present specific insights more clearly and effectively, as a participant noted, *"(Live Charts) clearly explain what the chart shows."* Users with animated charts also reinforced this fact by suggesting that *"Animated charts can sometimes make it more difficult to understand what is being shown, particularly when only using highlighting to emphasize certain aspects."* This observation may explain the disparity in the higher score for the *Insightful* dimension for Live Charts. Furthermore, integrated with the GPT-3 model, the narrator can analyze the data table and combine it with relevant news or context. As noted by one user with Live Charts, *"There's more context with narration allowing for greater insights into the data and increasing understanding."*

In addition to the benefits of animation and narration on their own, we have also received feedback highlighting the advantages of combining the two. Users with Live Charts reported that the interplay between animation and narration could enhance the overall experience. One participant in Live Charts noted that *"It is interesting hearing voice while looking at the chart equally."*

They also suggested that ensuring that the information presented in both the voice-over and the chart matches accurately is crucial. Furthermore, they rated the *Memorable* dimension significantly higher than participants with the static format, while there was no significant difference between those with animated charts and static charts.

Although Live Charts offer a more engaging experience by incorporating animations and narrations, they have certain limitations. Some users with animated charts and Live Charts commented that Live Charts take up more file space than a static image. One user noted that *"Internet connection and speeds could be a potential drawback,"* while another one reported that *"(Live Charts) need to have the computer power to display it and cannot just print on paper to display."* The main reasons for these difficulties can be traced to the characteristics of videos, which consist of a sequence of images. Additionally, some users with animated charts and Live Charts felt it was not easy to pinpoint the precise moment when a particular insight was discussed, while static charts can be understood at a glance. This can result in the need to replay the Live Charts multiple times to locate the desired information. Currently, commercial video players can add small tags to videos' timelines, which can assist users in locating specific content. Such players would be helpful in the playback of Live charts as well.

We also found varying individual preferences for data communication among users. Although Live Charts was appreciated by a large number of participants for its ability to emphasize particular data points and help absorb the information, some participants pointed out that Live Charts may have limitations in highlighting only certain information, which could be viewed as a drawback, as one noted that *"You are more likely to only focus on the information that the narrator is pointing out rather than looking at the chart as a whole for yourself."* Regarding the method of receiving information, individuals may also exhibit distinct preferences. While some participants enjoy taking an information journey through Live Charts, others prefer a less guided approach to understanding the data by themselves, as noted by one user who stated that *"Live Charts don't give people as much time to think on their own."* The pace of Live Charts can also be divergent. Some participants feel that *"(Live Charts) are being explained too slowly."*, while others find that *"Live Charts may also go through the information too quickly, and people may miss information."* Based on the feedback received, we anticipate the development of an authoring tool that would allow users to customize the creation of Live Charts by controlling the information division and the pace for different scenarios. However, as our research primarily deals with the automatic generation of Live Charts, these concerns are not within the scope of our paper. We will delve deeper into these insightful remarks in the Section V-B.

D. Expert Interview

We conducted a series of interviews with three expert users from diverse fields to evaluate the effectiveness of our automatic generation workflow. The first expert (E1) has over three years of experience as a data journalist for a digital-news firm. The second expert (E2), possessing a master's degree in visual communication, has over five years of experience. She demonstrates her prowess as a UI/UX designer in a globally acclaimed software firm. The third expert (E3) is a senior researcher with over seven years of experience specializing in animation for data visualization.

We used an online meeting system to conduct the interviews, which began with an introduction to Live Charts followed by three examples. We then probed the experts on the quality of Live Charts and their methods for creating such charts. By introducing our automatic workflow, we solicited feedback on its effectiveness. Finally, we asked how it could potentially benefit their work and future endeavors.

All three experts agreed that our Live Charts were of high quality. They agreed that the narration logic and overall animation design in Live Charts are appropriate. Specifically, E3 believed that the progressive introduction of information is well-suited for public users, and E2 thought that the animation effects effectively complement the visual design. E1 stated that *"We have previously utilized video formats in our articles, and Live Charts have achieved a similar level of quality to our past efforts."* The experts also offered suggestions for improving Live Charts, with E1 suggesting that the narration could be more expressive and conversational in tone, rather than sounding too formal and robotic.

Regarding our workflow for automatically generating Live Charts, all the experts appreciated it and found it helpful. They currently use different tools for creating such charts, with E1 adopting existing authoring tools in her company to generate videos, E2 relying on Adobe PR or AE for video editing, and E3 using D3 through coding. E1 mentioned that their company's tool generates videos from data, but it may have limitations in terms of chart style. In contrast, our method can handle charts with different styles, including those for online use. Additionally, E1 stressed the significance of data accuracy, which our method achieves through the use of models for extraction. The experts acknowledged the value of our workflow but also offered suggestions for improving it. While E1 and E2 suggested incorporating an extension into familiar tools to avoid switching between applications, E1 also raised concerns about data privacy within the company and the risks associated with using external extensions. Additionally, E1 and E3 recognized the potential for our workflow to transform data news into videos in the future, enabling them to consume data articles more efficiently while performing other tasks. As an expert in visualization, E3 expressed a preference for a human-AI collaborative approach, in which the user inputs simple insight hints and animation types, and AI assists in making the charts more compact.

V. Discussion

This section begins with two key lessons learned from the research, specifically related to the format of Live Charts and the generation of Live Charts. The limitations of our research are discussed as follows.

A. The Live Chart Format

Reducing cognitive burden is crucial as visualizations become more prevalent across diverse media. Traditionally, individuals relied on their knowledge and experience to understand charts. However, the advent of Live Charts can potentially transform this dynamic, offering a new method of information consumption. According to our evaluation results, Live Charts are enjoyable, tell insightful stories, and improve data understanding. We received positive feedback from multiple users affirming that Live Charts offer enhanced accessibility for individuals with visual impairments by providing a multi-sensory experience. Our adopted accessible narration method [61] has proven instrumental in enabling BLVIs to gain a comprehensive understanding of the data presented, even without visual perception. Moving forward, further advancements can be made by integrating non-speech audio representations of data [70], [71] and efficient navigation strategies [72], [73]. We believe the potential applications of Live Charts are vast, as they can enhance the way information is presented and communicated in various visualizations, including data articles and slideshows. First, incorporating Live Charts into data articles to replace static ones can transform storytelling, offering readers a more engaging and informative experience. Users can access the information conveyed in data news by playing the Live Charts. According to expert feedback, they do not need to be fully focused on the news while watching; instead, they can absorb the information while multitasking. Slideshows are another example of the application. As a substitute for static chart images, Live Charts can enliven slides by employing a multi-sensory method of conveying information. For instance, replacing static charts with Live Charts in teaching materials in educational settings can significantly enhance students' learning experiences. This learning process fosters a more profound comprehension of intricate lessons.

B. The Creation of Live Charts

We utilize an auto-reviving method that dynamically changes frames with narration guiding users through the information presented.

Auto-Generated Live Charts: We have proposed a GNN-based model for recognizing basic charts in SVG format. Compared with pixel-based charts, vector graphics own higher precision, enabling us to extract data with greater confidence. Our model shows promise in effectively handling three fundamental chart types compared with pixel-based methods (Section IV-B): bar charts, pie charts, and line charts. Going forward, we can generalize this model for more chart types [74]. For instance, we anticipate applying this model to scatterplots, which may pose challenges in pixel-based format due to the occlusions. Furthermore, existing in-depth analysis [75] underscores an opportunity to recover the embedded data within the scatterplot. Moreover, while our approach currently shows modest adaptability to infographics with embedded charts [76], [77], it carries the potential for broadening its applicability, encompassing the personalized design of infographics [78]. We also explored the use of GPT-3, a popular LLM, to generate text narration for our data visualizations. For one thing, by adopting a prompt chain containing various insights, the LLM can generate narrations that cater to our needs, deriving fluent and natural-sounding narrations with key insights. For another, the LLM's training on online news data can assist in inferring the underlying reasons behind certain insights. For instance, in the example of a line chart, as shown in Fig. 1(a1-a5), the LLM can infer that "The box office earnings in 2020 were significantly lower than in previous years due to the COVID-19 pandemic." This information may not be directly visible in the chart, but with the LLM's assistance, audiences can better understand the context and the story behind the data. Looking ahead, as LLMs continue to evolve and improve, they can potentially incorporate real-time information and help narrations remain relevant and timely.

Opportunities for Human-Machine Collaboration: Developing a fully auto-generated Live Chart that satisfies every user's needs and ensures the quality of the narration by GPT-3 [79] is an arduous task, if not impossible, as our user study confirms the challenge due to participants' diverse preferences for Live Chart styles. To address this issue, human-machine collaboration is a promising solution [80], [81]. We allow users to first select their preferred scenario, tone, and length with the assistant from the LLM. Our methods enable the generation of an initial draft of the chart that users can customize to align with their specific narration and animation requirements, reducing the time and effort. In addition, human involvement becomes crucial for assessing the quality of outcomes, given that LLMs are prone to producing inaccuracies [82]. Although existing methods, as discussed in Section III-B, are used to address errors, the rapid evolution of LLMs presents a challenge for automatic error detection as it is not controllable. Incorporating human judgment into the process maybe a valuable solution, as humans can quickly identify and correct errors at an early stage, avoiding unnecessary downstream efforts. Furthermore, there is an expectation for future models to include automated verification features. We also acknowledge the importance of on-the-fly collaboration, as highlighted by experts we interviewed. To facilitate this, we plan to integrate our functionality as extensions into existing tools, such as data processing applications (e.g., Excel or Tableau) and web browsers. This integration will address the inconvenience of switching between different tools. Users can export charts in diverse formats by incorporating our solution into data processing tools, facilitating data sharing and presentation across various platforms and devices. Meanwhile, integrating with web browsers will enable convenient conversion of online static charts for sharing or presentation purposes.

C. Limitations

Our research has limitations. First, we use predefined animations, which may result in limited diversity and somewhat repetitive visual effects. For example, our highlight animation effect utilized fixed colors without regard to the holistic color scheme. In the future, we intend to extend the animation library, ensuring greater diversity and context-aware visual effects. Second, the generalizability of our understanding model needs improvement. It performs well in dealing with SVGs generated by declarative and imperative language. However, challenges may arise when handling visualizations created with highly flexible tools (e.g., Adobe Illustrator)[8] or hand-coded approaches (e.g., Vue, React). Future research could focus on

[8] https://www.adobe.com/products/illustrator

refining the generalizability of our GNN-based approach to address diverse visualization structures. Third, the outcomes generated by GPT may include errors [83]. Despite the use of rule-based methods to minimize errors, these approaches are not exhaustive in resolving all issues [82]. Moving forward, we hope that improvements in LLMs will lead to more precise and high-quality answers. Fourth, we adopt a fixed narration outline in this work for a gradual narration. However, we acknowledge the limitations associated with this approach and propose to leave it as future work for LLMs to generate more adaptable outlines. Fifth, the participants in both the user study and the expert interview have limited coverage. While we have made efforts by enlisting individuals with varied backgrounds and experiences, we recognize certain limitations, such as the unequal gender distribution in the user study and the absence of specific types of experts in the expert interview. In the future, we hope to gain a more comprehensive understanding of how Live Charts support data presentation for a diverse user base in long-term, real-world applications, thereby addressing these limitations.

VI. Conclusion

In this paper, we have introduced Live Charts as a new format of data presentation that enhances traditional static charts by providing rich animations and audio narration. To automate the creation of Live Charts, we decompose complex information from static charts by adopting computer vision techniques and large natural language models. Our process first involves extracting data and visual encodings from the chart image, then generating data insights based on this information. We then create narrations and design animations, ultimately combining them to form a Live Chart. We conducted various assessments to evaluate the effectiveness of Live Charts and our auto-reviving process, including use cases, chart understanding performance test, a user study, and expert interviews. The evaluation results demonstrate that Live Charts provide a multi-sensory and engaging experience, with animations and narration significantly contributing to data comprehension. Leveraging GPT-4's multimodal input and enhanced performance compared to GPT-3, future research can delve into advanced techniques for automating chart comprehension through LLM and enriching presentations with more vivid narration and animation.

References

[1] W. Zhao, G. Wang, Z. Wang, L. Liu, X. Wei, and Y. Wu, "A uncertainty visual analytics approach for bus travel time," *Vis. Inform.*, vol. 6, no. 4, pp. 1–11, 2022.

[2] G. Zhang, Z. Zhu, S. Zhu, R. Liang, and G. Sun, "Towards a better understanding of the role of visualization in online learning: A review," *Vis. Inform.*, vol. 6, no. 4, pp. 22–33, 2022.

[3] R. Li et al., "Causality-based visual analysis of questionnaire responses," *IEEE Trans. Vis. Comput. Graphics*, vol. 30, no. 1, pp. 638–648, Jan. 2024.

[4] G. Shirato, N. Andrienko, and G. Andrienko, "Identifying, exploring, and interpreting time series shapes in multivariate time intervals," *Vis. Inform.*, vol. 7, no. 1, pp. 77–91, 2023.

[5] F. Amini, N. H. Riche, B. Lee, J. Leboe-McGowan, and P. Irani, "Hooked on data videos: Assessing the effect of animation and pictographs on viewer engagement," in *Proc. Int. Conf. Adv. Vis. Interfaces*, 2018, pp. 1–9.

[6] D. Ren, M. Brehmer, B. Lee, T. Hollerer, and E. K. Choe, "ChartAccent: Annotation for data-driven storytelling," in *Proc. IEEE Pacific Vis. Symp.*, 2017, pp. 230–239.

[7] F. Chevalier, N. H. Riche, C. Plaisant, A. Chalbi, and C. Hurter, "Animations 25 years later: New roles and opportunities," in *Proc. Int. Work. Conf. Adv. Vis. Interfaces*, 2016, pp. 280–287.

[8] D. Deng et al., "Revisiting the design patterns of composite visualizations," *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 12, pp. 5406–5421, Dec. 2023.

[9] Q. Wang, Z. Li, S. Fu, W. Cui, and H. Qu, "Narvis: Authoring narrative slideshows for introducing data visualization designs," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 1, pp. 779–788, Jan. 2019.

[10] Y. Wang, Y. Gao, R. Huang, W. Cui, H. Zhang, and D. Zhang, "Animated presentation of static infographics with infomotion," *Comput. Graph. Forum*, vol. 40, pp. 507–518, 2021.

[11] S. E. Kahou, V. Michalski, A. Atkinson, A. Kadar, A. Trischler, and Y. Bengio, "FigureQA: An annotated figure dataset for visual reasoning," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–20.

[12] C. Lai, Z. Lin, R. Jiang, Y. Han, C. Liu, and X. Yuan, "Automatic annotation synchronizing with textual description for visualization," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2020, pp. 1–13.

[13] A. Mahmood, I. S. Bajwa, and K. Qazi, "An automated approach for interpretation of statistical graphics," in *Proc. 6th Intell. Hum.-Mach. Syst. Cybern.*, 2014, pp. 376–379.

[14] A. Balaji, T. Ramanathan, and V. Sonathi, "Chart-Text: A fully automated chart image descriptor," 2018, *arXiv: 1812.10636*.

[15] A. Srinivasan, S. M. Drucker, A. Endert, and J. Stasko, "Augmenting visualizations with interactive data facts to facilitate interpretation and communication," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 1, pp. 672–681, Jan. 2019.

[16] K. Davila, S. Setlur, D. Doermann, B. U. Kota, and V. Govindaraju, "Chart mining: A survey of methods for automated chart analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 3799–3819, Nov. 2021.

[17] J. Gao, Y. Zhou, and K. E. Barner, "View: Visual information extraction widget for improving chart images accessibility," in *Proc. IEEE Int. Conf. Image Process.*, 2012, pp. 2865–2868.

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

[19] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer, "ReVision: Automated classification, analysis and redesign of chart images," in *Proc. Annu. ACM Symp. User Interface Softw. Technol.*, 2011, Art. no. 393.

[20] D. Masson, S. Malacria, D. Vogel, E. Lank, and G. Casiez, "ChartDetective: Easy and accurate interactive data extraction from complex vector charts," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2023, pp. 1–17.

[21] J. Harper and M. Agrawala, "Deconstructing and restyling D3 visualizations," in *Proc. Annu. Symp. User Interface Softw. Technol.*, 2014, pp. 253–262.

[22] J. Harper and M. Agrawala, "Converting basic D3 charts into reusable style templates," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 3, pp. 1274–1286, Mar. 2018.

[23] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 data-driven documents," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011.

[24] E. Hoque and M. Agrawala, "Searching the visual style and structure of D3 visualizations," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 1236–1245, Jan. 2020.

[25] D. H. Kim, E. Hoque, and M. Agrawala, "Answering questions about charts and generating visual explanations," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2020, pp. 1–13.

[26] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-Lite: A grammar of interactive graphics," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 341–350, Jan. 2017.

[27] H. Li, Y. Wang, A. Wu, H. Wei, and H. Qu, "Structure-aware visualization retrieval," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2022, pp. 1–14.

[28] K. Xiong, Z. Luo, S. Fu, Y. Wang, M. Xu, and Y. Wu, "Revealing the semantics of data wrangling scripts with COMANTICS," *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 1, pp. 117–127, Jan. 2023.

[29] K. Kafle, B. Price, S. Cohen, and C. Kanan, "DVQA: Understanding data visualizations via question answering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5648–5656.

[30] N. Methani, P. Ganguly, M. M. Khapra, and P. Kumar, "PlotQA: Reasoning over scientific plots," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2020, pp. 1516–1525.

[31] R. Chaudhry, S. Shekhar, U. Gupta, P. Maneriker, P. Bansal, and A. Joshi, "LEAF-QA: Locate, encode & attend for figure question answering," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2020, pp. 3501–3510.

[32] R. Reddy, R. Ramesh, A. Deshpande, and M. M. Khapra, “FigureNet : A deep learning model for question-answering on scientific plots,” in *Proc. Int. Joint Conf. Neural Netw.*, 2019, pp. 1–8.

[33] T. Munzner, *Visualization Analysis and Design*. CRC Press, 2014.

[34] B. Lee, R. H. Kazi, and G. Smith, “SketchStory: Telling more engaging stories with data through freeform sketching,” *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2416–2425, Dec. 2013.

[35] N. Kong and M. Agrawala, “Graphical overlays: Using layered elements to aid chart reading,” *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 12, pp. 2631–2638, Dec. 2012.

[36] C. Bryan, K.-L. Ma, and J. Woodring, “Temporal summary images: An approach to narrative visualization via interactive annotation generation and placement,” *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 511–520, Jan. 2017.

[37] M. Lu et al., “Interaction+: Interaction enhancement for web-based visualizations,” in *Proc. IEEE Pacific Vis. Symp.*, 2017, pp. 61–70.

[38] V. O. Mittal, J. D. Moore, G. Carenini, and S. Roth, “Describing complex charts in natural language: A caption generation system,” *Comput. Linguistics*, vol. 24, no. 3, pp. 431–467, 1998.

[39] D. Shi, X. Xu, F. Sun, Y. Shi, and N. Cao, “Calliope: Automatic visual data story generation from a spreadsheet,” *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 2, pp. 453–463, Feb. 2021.

[40] H. Li, L. Ying, H. Zhang, Y. Wu, H. Qu, and Y. Wang, “Notable: On-the-fly assistant for data storytelling in computational notebooks,” in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2023, pp. 1–16.

[41] S. Kantharaj et al., “Chart-to-Text: A large-scale benchmark for chart summarization,” in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 4005–4023.

[42] J. Obeid and E. Hoque, “Chart-to-Text: Generating natural language descriptions for charts by adapting the transformer model,” in *Proc. Int. Conf. Natural Lang. Gener.*, 2020, pp. 138–147.

[43] C. Liu, L. Xie, Y. Han, D. Wei, and X. Yuan, “AutoCaption: An approach to generate natural language description from visualization automatically,” in *Proc. IEEE Pacific Vis. Symp.*, 2020, pp. 191–195.

[44] F. Amini, N. H. Riche, B. Lee, C. Hurter, and P. Irani, “Understanding data videos: Looking at narrative visualization through the cinematography lens,” in *Proc. Annu. ACM Conf. Hum. Factors Comput. Syst.*, 2015, pp. 1459–1468.

[45] F. Amini, N. H. Riche, B. Lee, A. Monroy-Hernandez, and P. Irani, “Authoring data-driven videos with DataClips,” *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 501–510, Jan. 2017.

[46] D. Shi, F. Sun, X. Xu, X. Lan, D. Gotz, and N. Cao, “AutoClips: An automatic approach to video generation from data facts,” *Comput. Graph. Forum*, vol. 40, no. 3, pp. 495–505, 2021.

[47] M. Shin et al., “Roslingifier: Semi-automated storytelling for animated scatterplots,” *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 6, pp. 2980–2995, Jun. 2023.

[48] Y. Cao, J. L. E. Z. Chen, and H. Xia, “DataParticles: Block-based and language-oriented authoring of animated unit visualizations,” in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2023, pp. 808:1–808:15.

[49] J. R. Thompson, Z. Liu, and J. Stasko, “Data animator: Authoring expressive animated data graphics,” in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2021, pp. 1–18.

[50] Z. Liu et al., “Data illustrator: Augmenting vector design tools with lazy data binding for expressive visualization authoring,” in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2018, pp. 1–13.

[51] T. Ge, Y. Zhao, B. Lee, D. Ren, B. Chen, and Y. Wang, “Canis: A high-level language for data-driven chart animations,” *Comput. Graph. Forum*, vol. 39, no. 3, pp. 607–617, 2020.

[52] T. Ge, B. Lee, and Y. Wang, “CAST: Authoring data-driven chart animations,” in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2021, pp. 1–15.

[53] H. Cheng, J. Wang, Y. Wang, B. Lee, H. Zhang, and D. Zhang, “Investigating the role and interplay of narrations and animations in data videos,” *Comput. Graph. Forum*, vol. 41, pp. 527–539, 2022.

[54] X. Jiang, L. Liu, C. Shan, Y. Shen, X. Dong, and D. Li, “Recognizing vector graphics without rasterization,” in *Proc. 34th Annu. Conf. Neural Inf. Process. Syst.*, 2021, pp. 24 569–24 580.

[55] R. Shi, X. Jiang, C. Shan, Y. Wang, and D. Li, “RendNet: Unified 2D/3D recognizer with latent space rendering,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5398–5407.

[56] S. Rubab, L. Yu, J. Tang, and Y. Wu, “Exploring effective relationships between visual-audio channels in data visualization,” *J. Vis.*, vol. 26, no. 4, pp. 937–956, 2023.

[57] P. Ma, R. Ding, S. Han, and D. Zhang, “MetaInsight: Automatic discovery of structured knowledge for exploratory data analysis,” in *Proc. Int. Conf. Manage. Data*, 2021, pp. 1262–1274.

[58] R. Ding, S. Han, Y. Xu, H. Zhang, and D. Zhang, “QuickInsights: Quick and automatic discovery of insights from multi-dimensional data,” in *Proc. Int. Conf. Manage. Data*, 2019, pp. 317–332.

[59] D. Deng, A. Wu, H. Qu, and Y. Wu, “DashBot: Insight-driven dashboard generation based on deep reinforcement learning,” *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 1, pp. 690–700, Jan. 2023.

[60] OpenAI, “GPT Best Practices,” 2023. [Online]. Available: https://platform.openai.com/docs/guides/gpt-best-practices

[61] A. Lundgard and A. Satyanarayan, “Accessible visualization via natural language descriptions: A four-level model of semantic content,” *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 1, pp. 1073–1083, Jan. 2022.

[62] Y. Wang et al., “DataShot: Automatic generation of fact sheets from tabular data,” *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 895–905, Jan. 2020.

[63] J. Wei, S. Kim, H. Jung, and Y.-H. Kim, “Leveraging large language models to power chatbots for collecting user self-reported data,” in *Proc. ACM Human-Comput. Interact.*, 2024, vol. 8, pp. 1–35.

[64] M. Zong and B. Krishnamachari, “A survey on GPT-3,” 2022, *arXiv:2212.00857*.

[65] F. Huth, T. Blascheck, S. Koch, and T. Ertl, “Studies and design considerations for animated transitions between small-scale visualizations,” *J. Vis.*, vol. 26, no. 6, pp. 1421–1443, 2023.

[66] M. Contributors, “MMYOLO: OpenMMLab YOLO series toolbox and benchmark,” 2022. [Online]. Available: https://github.com/open-mmlab/mmyolo

[67] M. Fey and J. E. Lenssen, “Fast graph representation learning with PyTorch geometric,” 2019, *arXiv: 1903.02428*.

[68] A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” in *Proc. 32nd Annu. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

[69] YoloV8, “Ultralytics YOLOv8,” 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[70] M. N. Hoque, M. Ehtesham-Ul-Haque, N. Elmqvist, and S. M. Billah, “Accessible data representation with natural sound,” in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2023, pp. 1–19.

[71] A. Sharif, O. H. Wang, A. T. Muongchan, K. Reinecke, and J. O. Wobbrock, “VoxLens: Making online data visualizations accessible with an interactive javascript plug-in,” in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2022, pp. 1–19.

[72] J. Zong, C. Lee, A. Lundgard, J. Jang, D. Hajas, and A. Satyanarayan, “Rich screen reader experiences for accessible data visualization,” *Comput. Graphics Forum*, vol. 41, no. 3, pp. 15–27, 2022.

[73] J. R. Thompson, J. J. Martinez, A. Sarikaya, E. Cutrell, and B. Lee, “Chart reader: Accessible visualization experiences designed with screen reader users,” in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2023, pp. 1–18.

[74] L. Ying et al., “VAID: Indexing view designs in visual analytics system,” in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2024, pp. 1–15.

[75] A. Sarikaya and M. Gleicher, “Scatterplots: Tasks, data, and designs,” *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 402–412, Jan. 2018.

[76] L. Ying et al., “GlyphCreator: Towards example-based automatic generation of circular glyphs,” *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 1, pp. 400–410, Jan. 2022.

[77] J. Lin, Y. Cai, X. Wu, and J. Lu, “Graph-based information block detection in infographic with gestalt organization principles,” *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 3, pp. 1705–1718, Mar. 2023.

[78] L. Ying et al., “MetaGlyph: Automatic generation of metaphoric glyph-based visualization,” *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 1, pp. 331–341, Jan. 2023.

[79] S. R. Bowman, “Eight things to know about large language models,” 2023, *arXiv:2304.00612*.

[80] H. Li, Y. Wang, Q. V. Liao, and H. Qu, “Why is AI not a panacea for data workers? An interview study on human-AI collaboration in data storytelling,” 2023, *arXiv:2304.08366*.

[81] H. Li, Y. Wang, and H. Qu, “Where are we so far? Understanding data storytelling tools from the perspective of human-AI collaboration,” in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2024, pp. 1–19.

[82] Y. Ge et al., “OpenAGI: When LLM meets domain experts,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, vol. 36.

[83] Y. Tian et al., “ChartGPT: Leveraging LLMs to generate charts from abstract natural language,” *IEEE Trans. Vis. Comput. Graphics*, to be published, doi: 10.1109/TVCG.2024.3368621.

Lu Ying received the BEng degree in digital media technology from Zhejiang University. She is currently working toward the PhD degree with the State Key Lab of CAD&CG, Zhejiang University. Her main research interests are on data storytelling, glyph-based visualization. She is dedicated to integrating the AI technique into visualization to ease the creation of visualization. For more details, please refer to https://yiyinyinguu.github.io/.

Yun Wang is a senior researcher in the Data, Knowledge, Intelligence (DKI) Area with Microsoft Research Asia. Her research lies in the intersection of human-computer interaction (HCI), information visualization (VIS), artificial intelligence (AI), and data science (DS). Her work facilitates Human-Data Interaction, Human-AI Collaboration, and Data Storytelling through an interdisciplinary approach. For more details, please refer to https://www.microsoft.com/en-us/research/people/wangyun/.

Haotian Li received the BEng degree in computer engineering from HKUST. He is currently working toward the PhD degree in computer science and engineering with the Hong Kong University of Science and Technology (HKUST). His main research interests are data visualization, visual analytics, human-computer interaction and online education. For more details, please refer to https://haotian-li.com/.

Shuguang Dou is currently working toward the PhD degree with the College of Electronics and Information Engineering, Tongji University, Shanghai, China. His research interests include Infographic understanding, person re-identification, NAS benchmark, X-ray, and remote sensing. He has published papers in ICLR, AAAI, *IEEE Transactions on Image Processing*, *IEEE Transactions on Information Forensics and Security*, and IEEE Transactions on Circuits and Systems for Video Technology. He is a program member of ICML, NIPS, and ICLR. For more information, please visit https://shuguang-52.github.io/.

Haidong Zhang received the PhD degree in computer science from Peking University, China. He is currently a principal architect with Microsoft Research Asia. His research interests include visualization and human-computer interaction.

Xinyang Jiang received the BE degree from Zhejiang University in 2012 and the PhD degree from Zhejiang University in 2017. He is now a researcher from Microsoft Research Asia. Before joining MSRA, he was a researcher from Tencent Youtu Lab. His main research areas include cross-modal retrieval, computer vision, and pedestrian re-identification. He has published more than ten papers in CVPR, ICML, NIPS, ICLR, ECCV, AAAI, ACMMM, *IEEE Transactions on Image Processing* and other top conferences and journals on computer vision and artificial intelligence. He is a program member of AAAI, CVPR, MM and other conferences, and a reviewer for *IEEE Transactions on Circuits and Systems for Video Technology*, *IEEE Transactions on Image Processing* and other journals.

Huamin Qu (Member, IEEE) received the BS degree in mathematics from Xi'an Jiaotong University, China, the MS and PhD degrees in computer science from the Stony Brook University. He is a chair professor with the Department of Computer Science and Engineering (CSE), the Hong Kong University of Science and Technology (HKUST) and also the director of the interdisciplinary program office (IPO) of HKUST. 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). For more information, please visit http://huamin.org/.

Yingcai Wu (Senior Member, IEEE) received the PhD degree in computer science from The Hong Kong University of Science and Technology. He is a professor with the State Key Lab of CAD&CG, Zhejiang University. His main research interests are information visualization and visual analytics, with focuses on urban computing, sports science, immersive visualization, and social media analysis. Prior to his current position, he was a postdoctoral researcher with the University of California, Davis from 2010 to 2012, and a researcher with Microsoft Research Asia from 2012 to 2015. For more information, please visit http://www.ycwu.org.