SceneLoom: Communicating Data with Scene Context

Lin Gao, Leixian Shen, Yuheng Zhao, Jiexiang Lan, Huamin Qu, Siming Chen

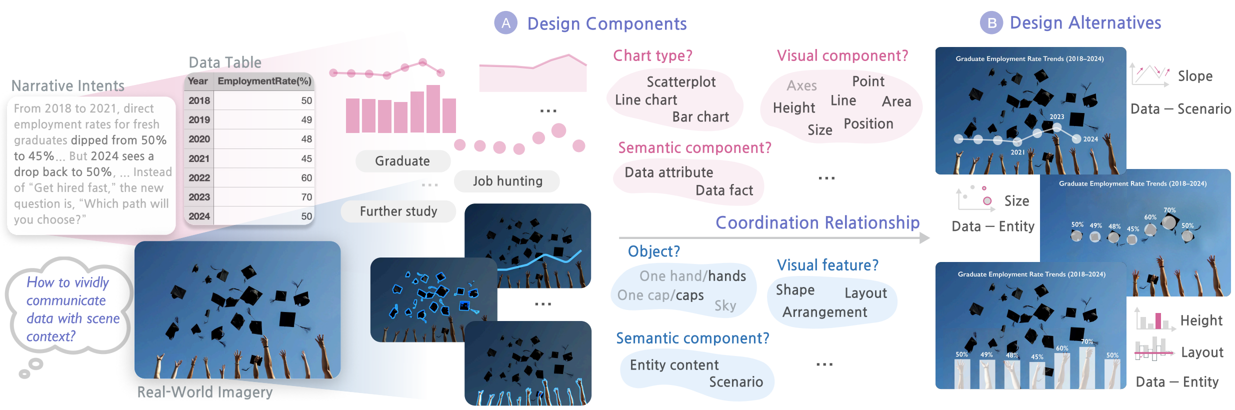

Fig. 1: SceneLoom explores creative ways to blend data visualization with real-world scene context for expressive data-driven storytelling. Given narrative intents and data, SceneLoom (A) bridges the complex and diverse design space between data visualization and scene imagery, and (B) generates contextually expressive design alternatives.

Abstract— In data-driven storytelling contexts such as data journalism and data videos, data visualizations are often presented alongside real-world imagery to support narrative context. However, these visualizations and contextual images typically remain separated, limiting their combined narrative expressiveness and engagement. Achieving this is challenging due to the need for fine-grained alignment and creative ideation. To address this, we present SceneLoom, a Vision-Language Model (VLM)-powered system that facilitates the coordination of data visualization with real-world imagery based on narrative intents. Through a formative study, we investigated the design space of coordination relationships between data visualization and real-world scenes from the perspectives of visual alignment and semantic coherence. Guided by the derived design considerations, SceneLoom leverages VLMs to extract visual and semantic features from scene images and data visualization, and perform design mapping through a reasoning process that incorporates spatial organization, shape similarity, layout consistency, and semantic binding. The system generates a set of contextually expressive, image-driven design alternatives that achieve coherent alignments across visual, semantic, and data dimensions. Users can explore these alternatives, select preferred mappings, and further refine the design through interactive adjustments and animated transitions to support expressive data communication. A user study and an example gallery validate SceneLoom's effectiveness in inspiring creative design and facilitating design externalization.

Index Terms—Creativity Support, Data Communication, Scene Context, Vision-Language Model

1 Introduction

In data-driven storytelling practices (*e.g.,* data journalism or data videos), real-world scenes and data visualizations serve as two foundational visual elements, each contributing in distinct yet complementary ways to the overall narrative [49]. Real-world scenes, *i.e.,* images or footage of environments, events, or activities, can provide spatial and temporal context [58], evoke emotional resonance [31], and offer visual cues that can inform the design of accompanying visualizations [10,25]. Meanwhile, in the data-driven storytelling context, data visualizations often serve to convert abstract information into graphical representations that highlight patterns and insights central to the narrative.

Although complementary, real-world scenes and data visualizations differ in fundamental ways. Real-world scenes often convey subjective narratives through concrete imagery, while data visualizations can encode abstract data relationships and emphasize factual clarity. As shown in Fig. 1, the scene depicts the celebratory moment of tossing graduation caps, conveying emotional and subjective intent, while the data visualization presents objective data on graduate employment. These modalities differ in information type, perceptual mode, and communicative goals, introducing distinct design considerations in areas such as semantics and visual features. Such differences make their seamless integration particularly challenging.

Recent studies have explored integrating real-world elements into data visualization as foreground objects or background canvases to enhance visual expressiveness [69]. These approaches either rely on data attribute mapping [81] or visual feature alignment [28], with many leveraging text-to-image models to reinforce semantic consistency [11,67]. However, these methods often ignore the systematic understanding of image content and data visualization, thus lacking fine-grained alignment. For instance, they tend to overlook critical structural components within visualizations, such as coordinate systems and spatial layouts, as well as the rich narrative contexts inherent in real-world scenes, like spatial relationships and character roles. Without a structured framework to analyze and align the expressive dimensions of both modalities, such approaches limit the space of design possibilities and constrain the system's ability to support open-ended exploration and creative design.

Therefore, coordinating real-world scenes with data visualizations remains a significant challenge. As noted earlier, the two modalities differ in the types of elements they contain and the design dimensions for effective coordination. Such divergence makes it inherently difficult

- L. Gao, Y. Zhao, J. Lan and S. Chen are with Fudan University. S. Chen is also with Ji Hua Laboratory and Shanghai Key Laboratory of Data Science. S. Chen is the corresponding author. E-mail: lingao23@m.fudan.edu.cn, simingchen@fudan.edu.cn.
- L. Shen and H. Qu are with the Hong Kong University of Science and Technology.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

to construct a systematic understanding of their components and the potential mappings between them, thereby limiting the space for creative design. Moreover, narrative-driven coordination goes beyond simply overlaying or juxtaposing visuals. The key challenge lies in simultaneously accounting for shared aspects (*e.g.,* spatial structures, thematic focus) and resolving inconsistencies (*e.g.,* mismatched element correspondences and conflicting narrative cues). Achieving this level of coordination requires a deep understanding of elements, their semantic roles, and spatial positioning, which is particularly challenging in the context of complex real-world scenes. Meanwhile, although Vision-Language Models (VLMs) excel at general visual understanding, they struggle to grasp the underlying design logic and contextual reasoning needed for effective coordination without additional knowledge.

To address these issues, we first conducted a formative study to analyze design components in data visualizations and real-world scenes and derive a set of coordination relationships from visual and semantic perspectives. Building on these insights, we developed SceneLoom, a prototype system that enables context-aware coordination between data visualizations and real-world scenes based on narrative intents, resulting in expressive and creative design outcomes. Given narrative text, a data table, and real-world imagery, SceneLoom begins with data preparation to extract relevant design components. The VLM-powered coordination process is structured in two stages: perception and reasoning. In the perception stage, components are specified along key dimensions of the design space to support VLM interpretation. In the reasoning stage, the VLM performs design mapping guided by a set of derived design considerations, including spatial organization, shape similarity, layout consistency, and semantic binding. To support fine-grained alignment, SceneLoom enables visualization adjustment and image editing to resolve data-element conflicts. After user refinement, the system further generates animated transitions for visualization elements to enhance narrative flow. We evaluated the system through a user study and curated an example gallery to demonstrate its expressive potential. Our main contributions are as follows:

- A design space that identifies design components and coordination relationships between real-world scenes and data visualizations to support visual alignment and semantic coherence.
- SceneLoom, a prototype system that supports context-aware coordination between real-world scenes and data visualizations. It integrates VLM-powered perception and reasoning to enable fine-grained alignment and creative support.
- An example gallery and a user study to validate the expressiveness and effectiveness of SceneLoom.

2 Related Work

This section reviews related work on blending data with real-world elements, image-driven creative tools, and VLM-based visual reasoning.

2.1 Blending Data Visualization with Real-World Elements

With growing attention to the physical world [20], as well as personal [22] and societal data [42], embedded visualization has expanded beyond Augmented Reality (AR) [66] into 2D contexts, enhancing links between real-world elements and data.

In 2D settings, recent studies have also explored ways to embed real-world elements or scenes into visual representations. For example, Infomage [11] integrates data visualizations into thematic images through image processing and visual distortion optimization. DataQuilt [81] extracts visual elements from raster images and binds them to data via an iterative process. Beyond Numbers [6] aligns scene elements with data through visual analogy, but departs from traditional charts in favor of naturalistic representations. With the rise of generative models [2, 45, 82], text-to-image techniques have further eased the integration of real-world elements. Several tools [28, 67, 69] extract visualization features and incorporate them as backgrounds or foreground inputs for generative models, using deep optimization and conditional generation to reduce uncertainty and improve control.

Despite progress, prior work mainly emphasizes visual presentation, often neglecting the narrative role of real-world scenes. While domain-specific applications, such as those in sports analytics [75, 88], enhance context understanding, they are limited by specific design spaces [74, 87]. The absence of a general design space hinders the diversity and expressiveness of integrated outcomes. We bridge this gap through a systematic analysis of modality components, identifying their core design elements and coordination relationships.

2.2 Creative Support Tools by Image-Driven Inspiration

Images are the multifaceted source of inspiration for designers and developers throughout ideation [10, 30], exploration [76], and prototyping [7]. Building on this, image-driven creativity-supporting tools are widely applied in graphic design, digital art, and storytelling.

Recent studies on reference images have primarily focused on visual and semantic aspects. Visual features, such as color palettes [54, 79], textures [38], and styles [84], are mapped more abstractly at a global level, shaping the overall perception of an image while conveying emotion. Geometric structures are mapped based on similarity and coherence. Chen *et al.* [86] identified compositional patterns in timeline infographics to inform new designs. Chilton *et al.* [8, 9] applied shape constraints to enable visual blending. Semantics serve not only as prompts for retrieval and generation but also as conceptual anchors, capturing entities and contexts that guide visual reinterpretation [10]. Moreover, combining multiple images introduces new dimensions of creativity. Like MetaMap [25], the relative positioning, shared features, and mapping of distances between images can spark unexpected ideas.

Regarding the mapping methods, Brickify [55] requires abstraction for free-form generation, and Data Pictorial [85] requires extracting precise element data to further input into generative models for next-step generation. With the advancement of LLMs, image-driven creativity has evolved to emphasize iterative refinement [83], alongside traditional stages such as brainstorming and alternatsive filtering.

While prior work highlights the creative potential of image-driven tools, most focus on open-ended mappings for exploratory or artistic use, with minimal design constraints. As a result, they often lack structured reasoning frameworks suited for goal-oriented, constrained scenarios such as data visualization. In our work, we focus on visualization contexts and examine which visual and semantic features in reference images can effectively inform design. By analyzing real-world elements across varying granularities and dimensions, we position images as sources of inspiration and carriers of narrative meaning.

2.3 VLM-driven Visual Understanding and Reasoning

VLMs extend Large Language Models (LLMs) with visual encoding, enabling models to "see" and perform tasks such as image captioning [18], visual question answering [68], and visual reasoning [36]. To enhance reasoning capabilities, recent works integrate traditional image processing [29] or deep learning methods [26, 33] to provide language-guided image tokens [62], supporting downstream tasks such as planning and tool execution [35, 64, 73].

As a special form of visual representation, data visualization presents unique challenges for VLMs [80]. Lundgard *et al.* [39] identified four levels of semantic understanding for data visualization: visual elements and properties, statistical concepts and relationships, graphical perception, and contextual or domain-specific insights. Recent work [23] has explored the ability of VLMs to understand charts across these levels in various downstream tasks. ChartInsighter [61] investigates VLMs' understanding of time series charts, focusing on the first two levels. Guo *et al.* [17] examine graphical perception tasks, such as position, height, and angle. Tasks requiring contextual understanding are often studied in domain-specific scenarios [37], while research on general storytelling focuses more on semantic coherence [34].

To support contextual understanding, we translate design space dimensions into structured specifications, enabling VLMs to better interpret task-relevant visual elements. While prior work, such as Meng *et al.* [41], has explored multi-image understanding in natural scenes, limited attention has been given to cross-modal relationships, particularly between data visualizations and natural images. To address this gap, we introduce a coordination process comprising perception and reasoning to guide VLMs in forming meaningful connections.

3 Formative Study

This section introduces the research question and our formative study to investigate it, as well as the derived design space and coordination strategies, which were further verified by design experts.

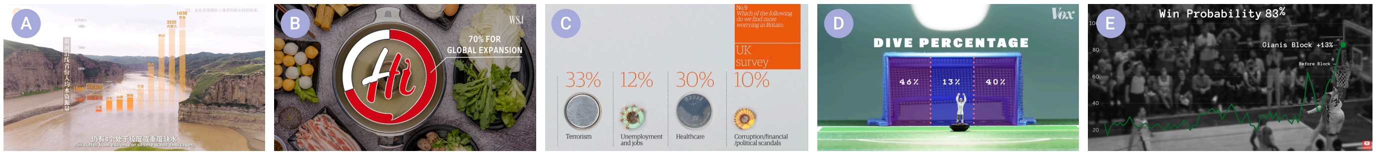

Fig. 2: Examples of integration cases. (A)Residential water resources along the Yellow River, using the riverbed as a baseline [71]. (B)Fund allocation with a pie chart matching the hot pot shape [24]. (C)Public concerns in the UK, represented by the size of physical objects [16]. (D)Goalkeeper dive percentages mapped to the goal layout [60]. (E)Win probability changes aligned with a key basketball dunk [40].

3.1 Research Question

Data visualizations and real-world scenes differ fundamentally in information type, perception modes, and communicative goals. This divergence creates tensions in their coordination: (1) semantic gaps between abstract data encoding and concrete scene semantics, and (2) perceptual competition when visual channels overlap. Therefore, these challenges lead to our central research question: *How to coordinate design components from data visualization and real-world imagery in a narrative-driven context?* To explore this, we focus on video-based storytelling, where temporal continuity and frame-by-frame structure reveal how visual elements evolve and interact with real-world scenes over time. Sequencing and camera motion enable the gradual introduction of elements, exposing transitions and coordination patterns. Building on this temporal structure, we analyze existing video cases to examine how data visualizations interact with real-world scenes (Sec. 3.2). In Sec. 3.3, we conducted frame-by-frame analysis to identify design components from both domains. Based on these findings, Sec. 3.4 presents coordination strategies that govern how these components are visually and semantically linked.

3.2 Corpus Analysis

To ground our research question in concrete examples and derive actionable insights for subsequent analysis, we conducted a corpus analysis of existing data videos. We first surveyed data videos from prior studies [56, 72], reputable news agencies (*e.g.,* Vox, BBC News), and major platforms (*e.g.,* YouTube, TikTok). Using keywords such as "data-driven stories" and "real-world data videos", we initially selected cases that featured tight and creative integration between data visualizations and real-world scenes. Due to limited cases, we broadened our scope to include videos on real-world topics with co-occurring visualizations, even if loosely coupled. In total, we selected 54 videos that featured both data visualizations and real-world elements.

These cases helped us extract common patterns of interaction and define a set of analytical dimensions. Guided by prior work [59, 66], we iteratively refined the coding scheme. Adopting an abductive coding approach, two authors independently coded the videos and resolved differences through discussion, resulting in six analytical dimensions. *Visualization components* capture how real-world elements are involved in the data graphics, categorized into coordinate systems (15), marks (26), and annotations (16). *Image components* describe real-world elements shown in the scene; due to their diversity, these were annotated in free-text form. *Compositional layout* captures whether the visualization appeared in the foreground (40) or background (14). *Visual inspiration* reflects perceptual connections such as shared shape (11), size (13), or position (11). *Semantic inspiration* refers to conceptual or thematic links, drawn from either metadata (31) or data context (23). *Narrative intent* refers to the communicative goal of the visualization-scene pairing, including explanation (26), comparison (20), and emphasis (9). These cases and coding results are available online[1]. Representative examples are shown in Fig. 2.

3.3 Identifying Design Components

Based on insights from the corpus analysis and related literature [55, 88], we analyze design components from two key aspects: visual cues and semantic content. These aspects often intertwine in practice, as shown in Fig.2A–E. Guided by this observation, we examine components from both the visualization side (Sec.3.3.1) and the real-world scene side (Sec.3.3.2). The resulting visual components are summarized in Fig.3.

[1]https://airtable.com/apparxcuOrUlTeKj3/shrFcnYr0QytfWLeE

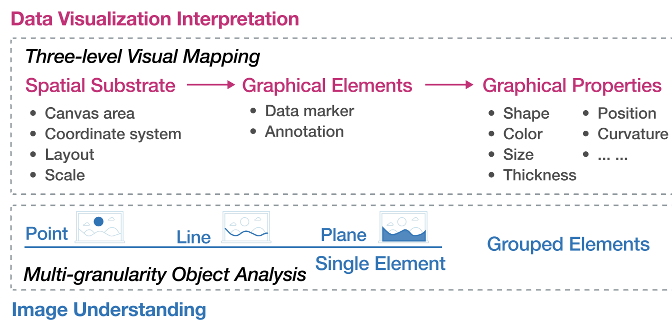

Fig. 3: Visual components in data visualization and real-world images.

3.3.1 Design Components from Data Visualization

In analyzing data visualization components, we identify visual components through a visual mapping framework [5] and characterize the semantic content conveyed by the data.

Visual Components. Previous work [51, 77, 86] has analyzed data visualizations from multiple perspectives, including form-related features. We build on the influential framework by Card *et al.* [5], which is rooted in visual perception and breaks down visualizations into three components: *Spatial Substrate*, *Graphical Elements*, and *Graphical Properties*. This framework forms the analytical foundation. Building on corpus insights, we extend it through a fine-grained analysis of components that support alignment with real-world scenes.

- *Spatial Substrate.* This refers to the foundational space where data visualizations are constructed. It encompasses data dimensions, axes, graphical boundaries, and spatial layout configurations. We categorize these as *canvas area*, *coordinate system*, *layout* and *scale*.
- *Graphical Elements.* These include geometric representations such as points, lines, and areas. In addition to *data markers*, they also include *annotations* such as gridlines and shaded regions.
- *Graphical Properties.* These mainly involve encoding channels, including *size*, *color*, and *shape*, to represent data dimensions or categories, or to emphasize specific aspects of the data. Different mark types involve distinct encoding properties. For example, line marks can be characterized by attributes such as *slope*.

Semantic Components. Data visualization conveys information and insights from the underlying data and its descriptive context. We refer to this as *Data Content*.

- *Data Content.* This includes data attributes, data values, and contextual information. As data visualizations are often shaped by user intent and narrative goals, we also consider data facts as part of the content.

3.3.2 Design Components from Real-world Scene

While computer vision has made progress in recognizing structural and semantic elements in real-world scenes, a unified classification for systematic analysis remains absent. Drawing on case studies and observations of complex scenes, we analyze visual composition and interpret meaning at multiple levels of granularity.

Visual Components. The gathered examples indicate that design components in real-world scenes exhibit varying levels of granularity, ranging from the entire scene as a background (Fig. 2A, E) to groups of elements representing data series (Fig. 2C), or individual elements (Fig. 2B, D). Accordingly, we adopt a multi-granularity perspective and categorize real-world design elements into two types: the *Single Element* and *Grouped Elements*.

- *Single Element.* This refers to an individual element, which can be a *point*, *line*, or *plane*. A point can represent a single physical object with no size constraints, but it does not represent a region, as the basketball in Fig. 2E. A line can be a physical line or a visually implied line formed by the arrangement of elements. In Fig. 2A, the visual line formed by the river surface is considered. A plane represents a defined region within the scene or can serve as a mask for an object, defining its boundaries or spatial extent. The examples in Fig. 2B and D all serve as the planes.
- *Grouped Elements.* These are combinations of multiple single elements organized with some logical coherence. Due to the diversity and complexity of grouping methods, our work primarily focuses on combinations where the shapes share similarities and have strong semantic associations. For instance, in Fig. 2C, the elements with similar circle shapes are considered together.

Semantic Components. For the semantic interpretation of real-world scenes, we should focus on the scene as a whole to convey the data context and the description of specific entities within the scene. So, we approach the semantic analysis from two angles:

- *Entity Objects.* We describe their physical or logical meanings within the narrative context, like the "hot pot", "coil", "basketball" and "goal" in Fig. 2.
- *Scenario.* We focus on elements such as time, location, events, and environment, which determine the interpretation of data and convey the background context. In Fig. 2A, the landscape of the Yellow River indicates the background of the water resource data.

3.4 Coordinating Design Components

Building on the identified design components, we outlined coordination strategies along two dimensions (visual alignment and semantic coherence) to guide the integration of data and scene in narrative context.

Visual Alignment. Building on the decomposition of visual components, we systematically mapped elements from real-world scenes to the data visualization space by considering spatial layout and intrinsic attributes (Fig. 4). In this three-level visual mapping process, real-world design components either directly serve as visualization elements or inform their generation based on contextual roles.

Point. In the *spatial substrate*, points typically serve as positional anchors, such as coordinate origins or canvas reference points to support spatial alignment. Within *graphical elements*, shape similarity enables points to map naturally onto point-based data markers. These may also serve as dot-like annotations in line charts or scatterplots to emphasize specific values. In Fig.2E, a basketball aligns with a highlighted data point through layout matching. Highlighting further utilizes the *properties* of points, such as color and size, as encoding channels. In Fig.2C, pole size variation visually encodes data magnitude.

Line. Lines are commonly used to represent connections, outlines, trends, and directional flows. In the *spatial substrate*, they may align with coordinate axes, serving as structural baselines, as illustrated in Fig. 2A. When layout consistency is maintained, lines naturally divide the canvas, requiring visualization designs to consider symmetry and potential transformations. Line length can also encode scale, influencing proportion and orientation. As *graphical elements*, lines appear as data markers, such as trend lines or reference lines that highlight baselines, thresholds, or comparisons. Their *graphical properties*, including slope, thickness, and color, help convey relationships and hierarchies. For example, overhead views of highways with varying widths can inspire area charts for traffic volume in network visualizations.

Plane. Planes support spatial alignment by defining canvas regions and shaping the boundaries within data. As *graphical elements*, they can serve as data markers illustrating distributions or shaded areas that categorize data subsets, as shown in Fig. 2B and D. Their *graphical properties*, such as shape and size, influence the structure of area charts and encode scale and magnitude. For example, the shape of the pot in Fig. 2B informs the design choice for the shape of the area chart.

In practice, visual alignment is primarily achieved through shape similarity and spatial correspondence between real-world elements and visualization components. The most direct way to support such alignment is through overlay or substitution, which allows data elements to be precisely positioned within the scene. These methods were also the most commonly observed in our corpus analysis and form the basis for deriving generalizable coordination principles.

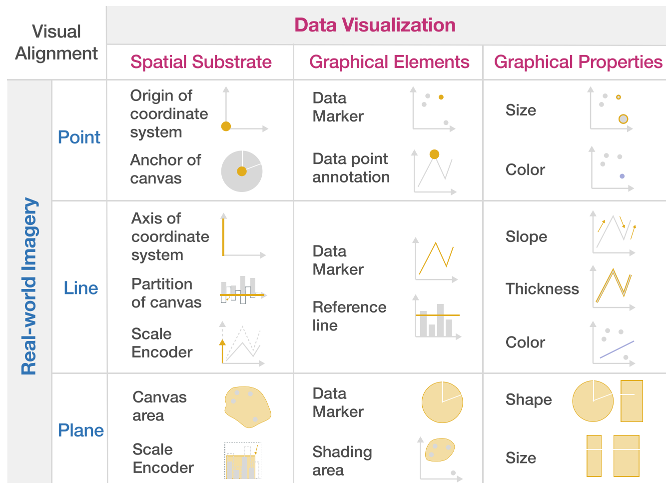

Fig. 4: Design space for visual alignment between data visualization and real-world imagery.

Semantic Coherence. To achieve semantic continuity, the integration of data visualizations and real-world scenes can be approached through two types of mappings:

Data Content – Entity Objects. Our case analysis reveals that semantic associations occur at varying levels. We categorize them into three types: *directly indicating data meaning*, *metaphorically representing data*, and *providing contextual information*. While the degree of association differs, all levels help bind data to objects, facilitating the instantiation of data concepts and enhancing the expression of data attributes. For example, in Fig. 2D, the goal area in the soccer field directly corresponds to numerical data (*e.g.,* goal distribution), serving as a visual representation of specific data columns. In contrast, Fig. 2A uses the shape of a river to guide the flow of a bar chart, offering contextual cues rather than directly encoding specific data points.

Data Content – Scenario. Contextual information in the scene complements data by situating it within real-world scenarios, emphasizing temporal, spatial, and contextual factors. Fig. 2E effectively leverages the moment of a slam dunk to complete the data narrative.

3.5 Expert Interview and Feedback

To validate the design space, we conducted 40-minute semi-structured interviews with two experienced information visualization experts. One (E1) has over seven years of experience in data journalism, and the other (E2) has five years of experience focused on creative visual communication. Both experts are external to the author team and offered insightful and constructive feedback. First, they strongly affirmed the significance of the research problem, noting that creatively aligning data visualizations with scene context is both meaningful and challenging. E1 highlighted that in her TV reporting work, *"it is often necessary to enhance content with data while preserving the authenticity of the scene,"* which fosters audience trust and emotional engagement.

Both experts noted that placing data visualizations in the foreground over real-world scenes is a common and effective design strategy. After reviewing our proposed design space, they agreed it offers a clear framework for linking scene elements with visual representations and found the identified combinations both valid and practical. E1 emphasized the importance of creativity, stating, *"In practice, many design ideas are inspired not only by the content itself but also by existing examples."* This highlights the value of presenting design alternatives through recommended templates to inspire users and support effective design decisions. Meanwhile, both experts are concerned about uncertainties in the design process, especially when scene images lack clear visual or semantic cues. *"In extreme cases"*, E1 noted, *"fallback strategies* *are needed."* E2 further stressed the need for interactive operations, *"Designers should be able to adjust and refine recommended results to improve readability and better meet their goals."* To support flexibility, we incorporate user manipulation features into our prototype system, enabling freeform creation and interactive refinement.

Based on this feedback, we further recognized the role of the design space in guiding design generation. In Sec. 4.3, we detail how data charts or images can be specified into actionable design representations and present key design considerations.

4 SceneLoom

In this section, we first give an overview of SceneLoom and then go through the workflow with an example in detail (Fig. 5). The interface and the outcomes of SceneLoom are illustrated in Sec. 4.5.

4.1 Workflow

To achieve visual alignment and semantic coherence during coordination, we propose a VLM-assisted workflow (Fig. 5) comprising data preparation, visual perception, and reasoning. The workflow takes structured data (CSV), narrative text, and real-world images (PNG/JPG) as input. In data preparation, SceneLoom extracts key narrative features (Fig. 5A), generate visualizations (Fig. 5B), and filter relevant images based on content and layout structure (Fig. 5C). Filtered elements are treated as design components, and VLMs extract their visual attributes using a standardized specification format (Fig. 5D-E). VLMs follow design considerations to guide the mapping process (Fig. 5F). To achieve fine-grained alignment, visualization adjustment is incorporated during reasoning (Fig. 5G), and final mappings are executed through LLM-driven tool invocation (Fig. 5H). Design alternatives are automatically evaluated by VLMs for data accuracy and visual communication effectiveness (Fig. 5I), enabling user selection. The interface also supports optional image editing to address inconsistencies (Fig. 5J). Once the design is selected, users can interactively refine the canvas (Fig. 5K), and SceneLoom continuously generates aligned animations (Fig. 5L).

Our workflow integrates state-of-the-art models to ensure the accuracy and robustness of outputs. Segment Anything Model (SAM) [26], Semantic-SAM [33], Holistically-Nested Edge Detection (HED) [70], and Mobile-Lite Structure Detector (M-LSD) [15] are used for image processing, and OpenAI's GPT-4o [1] supports visual understanding, reasoning, and code interpretation. Interactions with LLMs are implemented via natural language prompts that specify analysis goals, design constraints and generation tasks. Sample prompts and implementation details are provided in Appendix, which is included in the supplementary materials.

In following sections, we illustrate each stage using a case study from a 2024 U.S. survey on Christmas tree preferences[2]. The data highlights shifting consumer choices, including real trees, artificial trees, or no purchase. The narrative particularly emphasizes a growing preference for artificial trees over real ones.

4.2 Data Preparation

Given the complexity of image content and the diversity of data visualization, data preparation is essential. We adopt a narrative-driven approach in which the user's intent is first interpreted to uncover design-relevant cues. These cues then inform the generation of data visualizations and the selection of image elements.

Feature Extraction. Narrative intents play a central role throughout the process. As shown in Fig. 5B, SceneLoom extracts features such as data-related content, actions, and entity objects from the data table and input narration. In addition to identifying values and attributes, the system captures data facts to inform appropriate visualization mappings (*e.g.,* trends, comparisons, or distributions). Actions, particularly enter and emphasis, inform animation design by specifying element appearance and narrative focus. Concrete entity mentions (*e.g.,* "artificial Christmas trees", "Americans", and "a real one") are also identified automatically to support the following tasks, such as image filtering and matching. The extraction process is conducted through prompt-based queries, and the outputs are further normalized into structured forms. More details can be found in the supplementary materials.

Data Visualization Alternatives. Based on the uploaded data table and the data-related information extracted from the narrative intents, we expect LLMs to propose feasible designs and provide mapping interfaces to generate charts based on D3.js [3]. D3.js enables flexible generation of chart variants, and we provide a set of predefined rendering templates as callable interfaces. These visualization alternatives are stored in SVG format for subsequent operations and in PNG format to facilitate visual feature extraction by VLMs.

Filtering. We apply a filtering process to reduce the noise and cognitive load introduced by SAM, while preserving a diverse set of elements for flexible composition. Semantic-SAM provides coarse-grained labels, allowing the removal of segments unrelated to the narrative theme. Structured shapes are extracted by HED and M-LSD to retain visual distinction. Segmented elements are grouped by semantic and contour similarity into unified design components. For instance, in Fig. 5C, the two trees are merged by recognizing the "pine tree" semantics. The resulting segments, semantic labels, and preprocessing parameters are then passed to the sequential design mapping.

4.3 Visual Perception

To bridge human perception with AI understanding, we encode key design space dimensions into structured specifications. These specifications help the model reason about each element's role and provide a consistent reference for subsequent design mapping (Fig. 6).

Data Visualization Interpretation and Specification. The perception stage inputs both annotated SVGs and corresponding PNGs renderings into the VLMs. The SVGs provide structural and data-specific details, while the PNGs convey visual features such as shape, color, and overall appearance. As discussed in Sec. 3.3.1, data visualizations can be interpreted through a three-level visual mapping framework. Building on this framework and integrating data semantics, we propose a declarative specification (Fig. 6A) for describing visualization design elements. The specification begins by defining the chart type, followed by the corresponding visual mappings. In the spatial substrate section, the data fields are mapped to spatial axes. To accommodate layout diversity within the same chart type, this level also explicitly specifies the layout variant. In the graphical elements section, the focus is on geometric shapes used (*e.g.,* bar, line, circle) and their associated functional roles (*e.g.,* data marker, annotation). Each graphical element is represented as a pair: the element type combined with a natural language description of its encoding channel. A holistic, insight-driven perspective is adopted when interpreting visualizations. Visual insights are treated as semantic annotations and aggregated visual patterns that support narrative interpretation.

Image Element Understanding and Specification. To perceive image design components, the input includes the original image, extracted element masks, and preprocessed features such as detected lines and shapes. As shown in Fig. 6B and C, the specification describes design elements along two dimensions: granularity, distinguishing between individual and grouped elements; and element-level features, capturing geometric and semantic properties. For grouped elements, spatial arrangement and collective geometry are explicitly encoded (Fig. 6C). At the element level, each component is classified by geometric type, layout pattern, and semantic role.

4.4 Reasoning and Mapping

The mapping process is image-driven. Given an image inspiration, VLMs explore possible data visualizations through mapping reasoning (Fig. 5G–I). This involves design-based reasoning, visualization adjustments, image processing, and evaluation of design alternatives.

4.4.1 Design Mapping

Since the design space is organized from a bottom-up perspective, providing guiding principles for VLM understanding helps establish meaningful relationships between fundamental elements. Therefore, we introduce four key considerations through structured prompts (*spatial organization*, *shape similarity*, *layout consistency*, and *semantic binding*) to guide the construction of mapping relationships. By prompting with these considerations, VLMs are encouraged to reason in a

[2] https://www.nationalgeographic.com/environment/article/history-origin-artificial-Christmas-trees

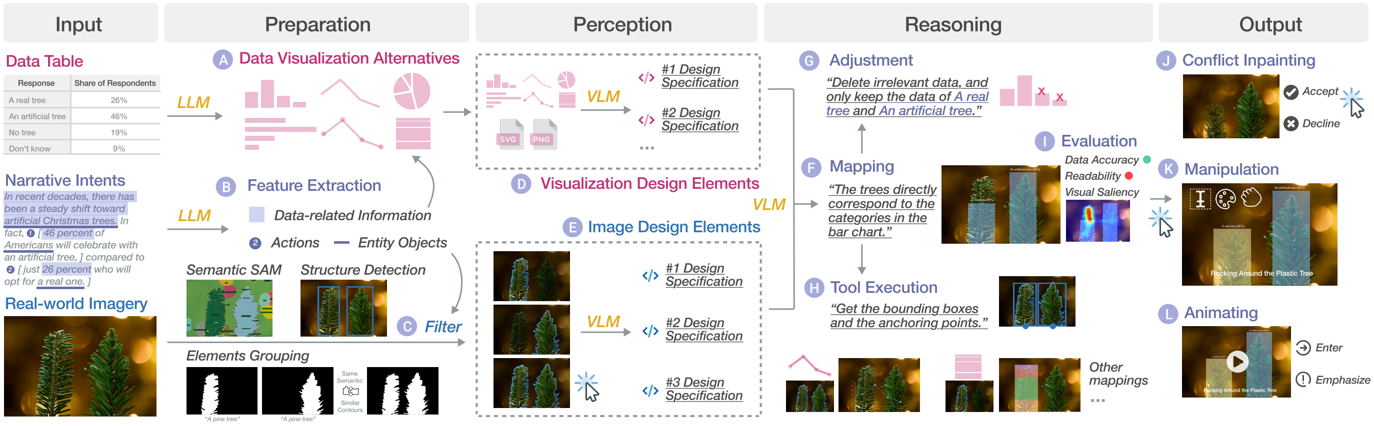

Fig. 5: The SceneLoom workflow for coordinating real-world imagery and data visualization based on narrative intent. It consists of five stages: Input, Preparation, Perception, Reasoning, and Output. A Christmas tree preference survey is used as an example to demonstrate the process.

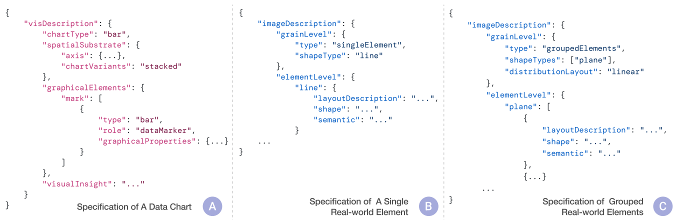

Fig. 6: Examples of design component specification. (A) A stacked histogram representing the distribution of tree cover. (B) A single bridge. (C) A combination of two Christmas trees.

Chain-of-Thought (CoT) manner [65], which is critical for decomposing complex alignment tasks into interpretable, step-by-step decisions.

First, ***spatial organization*** *determines how elements in the scene correspond to visual marks.* A single real-world object may map to an individual data point or a group of related points, while the overall scene may serve as the canvas or coordinate system. For grouped elements, a clear data-binding relationship is essential for meaningful mapping. ***Shape similarity*** *plays a crucial role in making visualizations intuitive.* Real-world objects should resemble the shapes used in data marks, such as lines in the scene matching line charts, and circular objects aligning with pie charts. Beyond basic shapes, finer details of shape features can also reflect data attributes. ***Layout consistency*** *ensures that the arrangement of elements in the scene mirrors the visualization structure.* The relative positions of individual objects should align with key points in the visualization, such as axes or reference lines. Meanwhile, the overall distribution of grouped elements (*e.g.,* scattered, clustered, *etc.*) should match patterns in the data to maintain a coherent spatial relationship. Finally, ***semantic binding*** *ties meaning to visualization.* Real-world objects should carry direct or metaphorical significance, linking their inherent qualities to data values or categories. Narrative elements in the scene, such as symbolic objects or contextual details, can further enhance this connection, making the visualization accurate and engaging. Additionally, an effective design plan should balance semantic relevance and visual alignment. While multiple forms of visual alignment are desirable, they are not strictly required to coexist within a single plan. The model is also expected to suggest potential improvements to better fulfill the intended design goals.

4.4.2 Visualization Adjustment

To ensure design coherence while minimizing changes to the original image, we constrain the model to adjust only the visualization. These adjustments preserve the underlying data and operate at two levels.

Data-level Adjustment. The model may perform data binding operations to better align image elements with corresponding data markers by *filtering* data irrelevant to the narrative. For instance, as illustrated in Fig. 5G, the model removes data entries such as "No tree" and "Don't know" to better match the remaining values with the two Christmas tree objects. In cases where prominent data insights are present, the model may further apply *classification* or *sorting* strategies to enhance the clarity of visual correspondence. The line chart in Fig. 7B3 sorts the data to fit the contour of the sky. The LLM receives both the dataset and a predefined code template, which allows it to perform data transformations and modify the visualization generation.

View-level Adjustment. This part is intended for visual alignment with the image, requiring operations such as *scale*, *translation*, or *rotation*. Individual visualization elements (*e.g.,* a data marker) or the entire visualization are processed as graphical objects. These operations rely on specific image processing parameters, which will be detailed in the next section.

4.4.3 Tool Execution

Inspired by Wang *et al.* [64], the model generates not only design and adjustment strategies but also corresponding tool interfaces and implementation parameters. These are specified through structured prompts and passed internally between system components to support automated operations. The relevant tools and parameters are listed in the Appendix. They are primarily designed for manipulating SVG elements within the visualization, including accessing specific elements, managing hierarchical relationships, adjusting element size, position, and rotation angle, and aligning these elements with counterparts in real-world scenes based on various alignment strategies. In addition, all image processing parameters(*e.g.,* bounding boxes, anchoring points, rotation angles, *etc.*) are accessible to facilitate tool execution.

4.4.4 Design Evaluation

The system evaluates design alternatives from data and visual perspectives, presenting data accuracy, visual readability, and attention analysis to assist user refinement. For data optimization, it not only verifies accuracy against data tables and bindings, but also detects and resolves conflicts such as encoding inconsistencies and misalignments between data and visual elements. For example, Fig.7J shows a height-encoding conflict, while Fig.1B highlights mismatches like a hat or hand misaligned with employment data. These issues are addressed through inpainting [78], reusing and repositioning content when possible, or using semantic-guided generation when necessary. Visual optimization consists of two components: visual readability and attention analysis. Readability is evaluated qualitatively, with the LLM providing feedback on factors such as the presence of occlusion, color distinction, and layout clarity , which are presented to users to support design decisions. Attention analysis uses saliency maps [21, 57] to simulate eye-tracking and identify visually prominent regions.

4.5 Interface

In Fig. 7, the interactive interface of SceneLoom comprises three views: *Input View* for data entry, *Creation View* for presenting and selecting design solutions, and *Editor View* for refining and rendering final output.

Input View. The left panel (Fig. 7A) enables users to upload multi-modal input materials, including data tables (Fig. 7A1), narrative intents

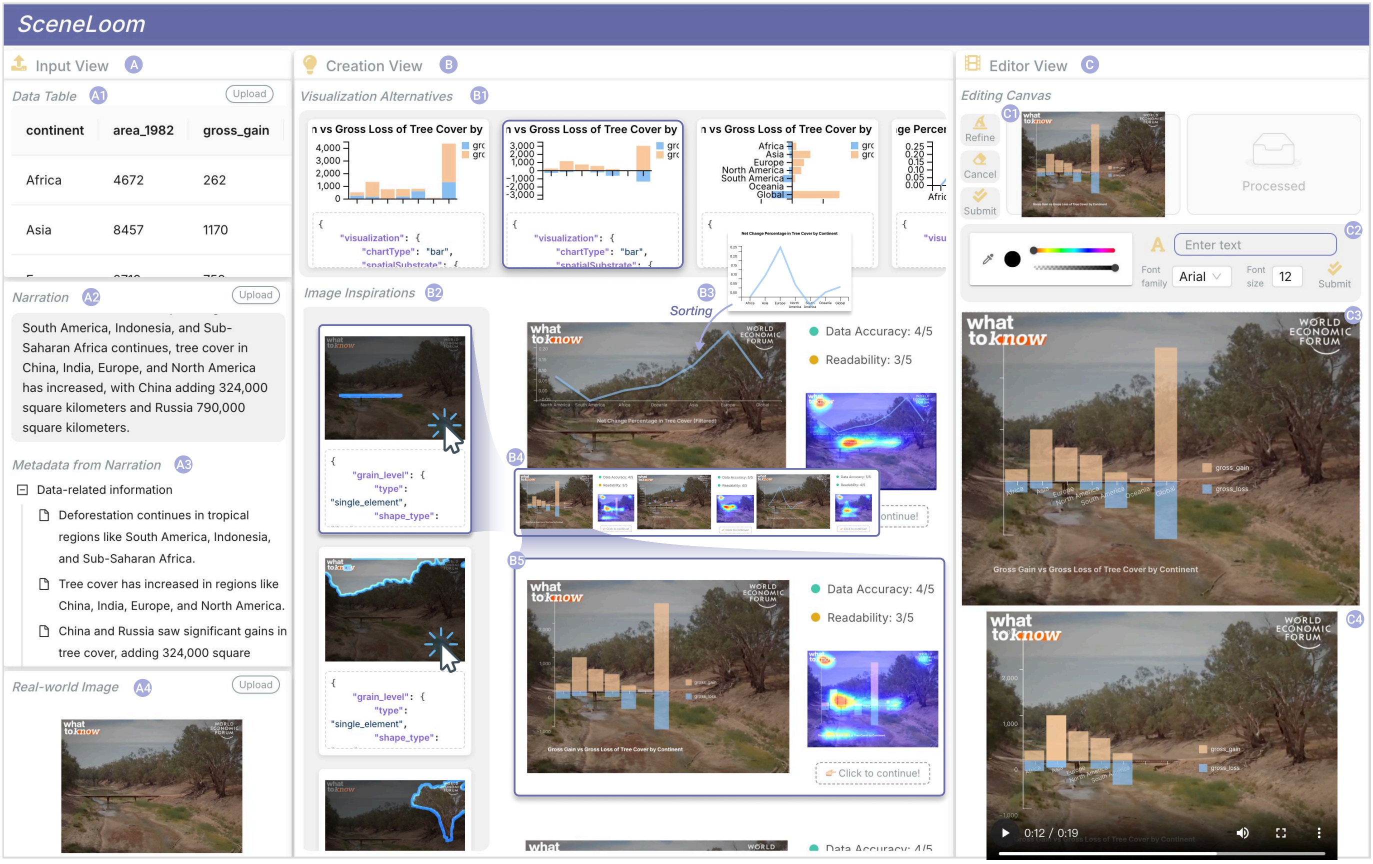

Fig. 7: SceneLoom interface within the example of global tree cover change implemented in our user study. After uploading the raw materials (A), users receive a series of inspirations from data visualizations and images. The system supports image-driven browsing and exploration of multiple design alternatives (B). After selecting a design alternative, the system supports fine-tuning and animation generation (C).

(Fig. 7A2), and real-world images (Fig. 7A4). Once the data tables and narrative intents are uploaded, the system extracts relevant information and displays it using a structured, tree-view interface (Fig. 7A3).

Creation View. Inspired by VIZITCARDS [19], all design elements are presented as design cards. A diverse collection of data visualization cards constitutes the *Visualization Alternatives* (Fig. 7B1), while highlighted visual elements extracted from the image form the *Image Inspirations* (Fig. 7B2). These cards are displayed in a scrollable panel layout, with each card encoded using structured specifications aligned with dimensions from our design space. Users are encouraged to explore design possibilities guided by visual cues. All visualization alternatives associated with a selected inspiration are spatially juxtaposed for comparison (Fig. 7B4). Each alternative, as illustrated in Fig. 7B5, presents a design template and evaluation metrics to support user decision-making, including data expression accuracy, readability, and visual saliency. Upon reviewing the options, users can select a preferred design and proceed by confirming their choice.

Editor View. This view supports the refinement of the selected design alternative and the creation of animations. Automated tools (Fig. 7C1) assist in identifying and resolving conflicts between image content and overlaid data through removal and inpainting operations. This step is optional; users seeking to preserve image authenticity may proceed directly to canvas editing. The canvas editor (Fig.7C2) allows for fine-grained control over visual elements, including dragging, color adjustment, scaling, rotation, and text insertion. Once editing is complete, SceneLoom generates an animation based on narrative-intent-driven actions and renders a dynamic visualization (Fig.7C3), which can be previewed and downloaded automatically.

5 Evaluation

We presented an example gallery and conducted a user study to validate the usability and effectiveness of SceneLoom. All examples discussed in this section were created by participants during the user study. Detailed evaluation results are provided in the Appendix.

5.1 Example Gallery

To demonstrate the expressiveness and appeal of the outcomes produced by users through SceneLoom, we collected design artifacts from participants in the user study, and a selection of them is shown in Fig. 8. Additional design outcomes, along with their animated versions, are provided in the supplementary materials. The example gallery includes different data themes, including economic, social, cultural, *etc.* We also present different design solutions based on the same materials (Fig. 8A, C), as well as design solutions generated from the same data table and narrative intents but with different image inputs (Fig. 8B). Additional examples are provided in the supplementary materials.

5.2 User Study

5.2.1 Experimental Set-up

Participants. We recruited 10 participants (P1-P10) interested in SceneLoom from various fields, including data analysts, graphic designers, journalists, and researchers in HCI or VIS. They were all between 20 and 35, including 6 females and 4 males. We first collected their basic information, through self-report (1 = No experience, 5 = Expert), to understand their expertise in data visualization (M=3.80, SD=1.03), visual design (M=3.40, SD=0.84), and video editing (M=3.40, SD=1.17). Each participant received a \$30 gift card after the study.

Procedure. By presenting several representative examples from the corpus, we introduced the purpose and procedure of the study. Each participant participated in an individual, in-person session, during which they were encouraged to adopt a think-aloud approach to verbalize their thought process. Firstly, participants were provided with datasets covering 10 topics and three to four relevant real-world images. They were asked to select two sets as the basis for their creative task, either out of interest in a particular topic or inspired by one of the images. Participants were then given 5–10 minutes to familiarize themselves

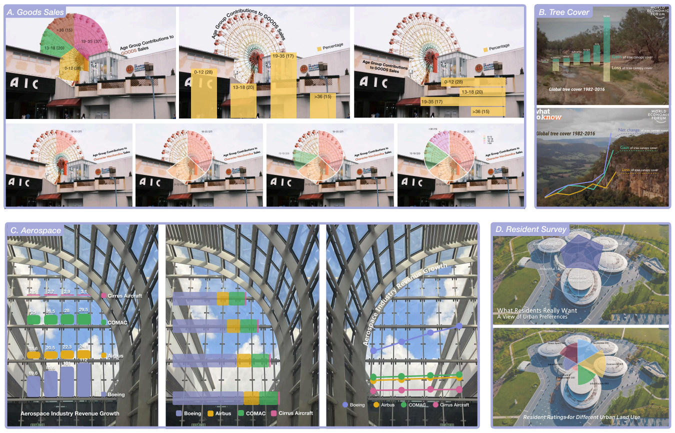

Fig. 8: Examples of design outcomes from our user study. (A) Top: Three designs created by P9 to express the relationship between goods sales and age. Bottom: Animation frames generated by the system. (B) A composition created by P4 to narrate global tree cover change using imagery from multiple sources. (C) A design by P8 illustrating the quarterly revenues of four airline companies. (D) A visualization created by P7 reflecting feedback on residents' preferences. These design examples are included in the supplementary materials.

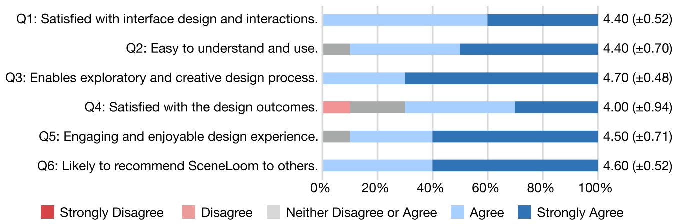

Fig. 9: Detailed subjective questions and corresponding user rating results. Q1 and Q2 assess system usability, Q3 and Q4 evaluate effectiveness, and Q5 and Q6 pertain to user recommendations. A 5-point Likert scale was employed to quantify user satisfaction, where a score of 5 represents strong agreement.

with the materials. During this time, we asked them about their initial creative ideas and encouraged them to articulate potential challenges. These ideas were primarily conveyed through verbal descriptions or sketches. Next, we asked participants to use our system to enable this creative process. We introduced the system's tutorial and helped them use the system with an example. In the whole task, participants would independently use SceneLoom to create and implement the selected set to complete the creation within approximately 30 minutes. We documented their findings and issues during the task and, with their consent, saved the final results. Finally, participants completed an exit questionnaire and participated in a semi-structured interview. Each session lasted 40–50 minutes.

5.2.2 Results and Analysis

Quantitative results of the questionnaire. All users completed the assigned tasks and provided feedback on SceneLoom, as shown in Fig. 9. We evaluated the system using a 5-point Likert scale from the perspectives of usability, effectiveness, and recommendability. The system design (M=4.40, SD=0.52) and user experience (M=4.40, SD=0.70) received positive feedback. Throughout the experiment, participants expressed a strong desire for exploration and demonstrated rich design ideas (M=4.70, SD=0.48). For the presentation of the final results (M=4.00, SD=0.94), some participants noted that the outcomes required further refinement. However, others reported that the intended design concepts were effectively communicated, and the need for post-adjustments did not significantly affect their overall perception. The overall design experience received positive feedback (M=4.50, SD=0.71), and participants were willing to recommend SceneLoom to others or use it in the future (M=4.60, SD=0.52).

The overall workflow effectively facilitated the generation and refinement of creative designs. Participants reported that SceneLoom provided substantial support at each stage of our workflow. P9 praised the effectiveness of data preparation. She remarked regarding the extraction of narrative intent, *"This structured representation of narrative intent allowed me to verify key information, avoiding omissions. It also enabled me to stay narrative-driven throughout the design process."* Participants emphasized the importance of integrating both data visualizations and real-world imagery during feature extraction. Some (4/10) reported that they often relied solely on one modality, either due to data complexity or visual bias, which sometimes resulted in missed design opportunities. P7 noted that the abstract explanations increased cognitive load, suggesting that natural language guidance might offer clearer support. Conversely, P2 appreciated the structured form, noting that *"it allowed me to align design suggestions with underlying principles, thereby enhancing the credibility of the generated outputs"*. This feedback highlights the need for more adaptive explanation strategies. Participants responded positively to the diversity and creativity of the design drafts produced by SceneLoom. These outputs validated initial ideas and inspired new directions, making the design process more exploratory. P4 praised the system's adaptability, sharing, *"I hadn't thought about reordering the data to match the contour features in the image... I'll definitely use this again to resolve mismatches."* Overall, participants expressed high satisfaction with the outcomes. P8 remarked that *"SceneLoom's recommendations exceeded my original ideas."* The integration of animation further contributed to a cohesive and polished presentation. Although P9 initially raised concerns about visual occlusion, she later found that the sequential animation effectively balanced content visibility with narrative flow (see Fig. 8A).

Design externalizations support creative ideation and assist in evaluating, selecting, and refining design solutions. Among the participants, some (2/10) had limited experience with data visualization, and some (3/10) were less proficient in extracting visual inspiration from images. After using SceneLoom, most participants (8/10) remarked that having abstract data represented through diverse visualizations allowed them to intuitively grasp distribution patterns and spatial configurations, as the visual forms made even simple datasets more interpretable and revealing. As P6 noted, *"Without the data visualization alternatives, I felt lost in the design process and could only form a vague idea."* Multi-faceted data often makes it challenging to establish meaningful connections to visual design. For instance, one of the examples featured quarterly production data from four aircraft manufacturers. P2 said, *"Even though the aircraft production data only had a few columns, I found it hard to see the temporal trends or differences between brands just by looking at numbers. SceneLoom enabled me to better comprehend how visual forms can represent such data more clearly."* Moreover, presenting the design draft gave participants useful references for making informed choices, evaluating alternatives, and iterating their ideas. For instance, in the case of economic goods (Fig. 8A), P9 initially intended to use railings as a metaphor for data mapping. However, after viewing the generated design inspired by railings, she reflected, *"The combination didn't work as well as I had expected – the visualization appeared abrupt in the scene."* This led her to reconsider and ultimately discard that design direction. Similarly, P5, who chose to represent a pie chart using the Ferris wheel structure, realized that its placement interfered with foreground buildings. Drawing on SceneLoom's readability suggestions, she refined the layout by reusing previously extracted building elements and layering them over the pie chart, thereby preserving the original visual depth of the scene.

Diverse design solutions reduced cognitive fixation and provided practical options for different storytelling needs. Most of the participants (9/10) reported that they often fell into fixed design patterns when encountering data or images, limiting their creative thinking. This rigidity was partly due to an overemphasis on dominant visual elements. For example, P1 remarked, *"I was too focused on the Ferris wheel and overlooked the surrounding details. I liked this design—it felt like a subtle but clever idea and brought an element of surprise."* P8 echoed this view, noting that SceneLoom's extraction of diverse visual features helped them move beyond dominant elements and discover overlooked but meaningful details. Design fixation was also evident in participants' visualization preferences. As P9 explained, *"If not for the diverse visualization choices, I probably would have defaulted to* *traditional bar charts without considering alternative layouts."* She continued to realize that such variation can significantly impact how the visualization integrates with the real-world scene. P4 noted that SceneLoom effectively mapped data to relevant objects using insightful, context-aware strategies in the tree cover case (Fig. 8B), making the process both engaging and valuable. P6 added that design diversity supports different communication goals. For example, some layouts suit formal, fact-based narratives, while others are better for general presentations, education, or storytelling.

Analysis of failure cases. Among the 16 participants, three did not initially receive design suggestions from SceneLoom, and one noticed that an obvious design pattern was not identified. Nonetheless, they remained patient and were willing to try again. We analyzed these cases and summarized the following reasons: (1) Simple narrative intents and insufficient semantic information limit effective filtering and reference for image elements, reducing inspiration. For example, when a user described *"optimistic oil consumption year by year"*, SceneLoom struggled to match this with relevant elements as no "oil" exists in the image. (2) High image complexity , characterized by dense visual elements, layered spatial structures, and intricate textures (*e.g.,* urban streetscape, busy indoor scenes), complicates segmentation and overlay alignment. Conversely, simple images also made it hard to identify suitable objects. In such cases, we suggested users either replace the image or apply a basic overlay approach. Apart from missing content, two design errors resulted from misalignment. While the design plans were correct and valid, irregular bounding boxes and occlusions caused positioning deviations, requiring manual adjustments. These common computer vision challenges were amplified in our scenario, which demanded both semantic and visual accuracy. Future work could further enhance the natural language understanding and object detection abilities of the system by tracking the most advanced models.

6 Discussion

In this session, we revisit our coordination method and system to discuss current limitations and key opportunities for advancing its adaptability, expressiveness, and user engagement.

Towards more expressive coordination between data visualization and real-world contexts. Our work presents a foundational approach for coordination while maintaining visual consistency and semantic coherence. While our corpus analysis focused on video-based storytelling, discussions with experts and users suggested that the proposed design space may generalize to other formats, such as interactive articles and scrollytelling. Future work will explore its adaptability across media to assess broader applicability and medium-specific considerations. Layout strategies like separation with curtain-opening effect, juxtaposition, and background substitution enable diverse narrative expressions. Each entails unique design and technical demands, from scene segmentation to rendering workflows. In narrative-driven contexts, additional factors often shape coordination. Aesthetic principles [27] such as contrast, hierarchy, and balance can inform the evaluation and refinement of design alternatives. Emotional cues [31] also play a crucial role in enhancing audience engagement and may guide the mapping process. We also observed that different coordination strategies influence the type and timing of animations, opening opportunities for motion design guided by narrative intent. Looking ahead, we envision extending this coordination framework to AR environments, which offer more immersive and spatially anchored storytelling [66].

Toward a more flexible and scalable workflow. Our current workflow requires users to provide diverse and multimodal input data. While this enables a clearer understanding of design requirements, it also introduces considerable challenges in data collection and preprocessing. To address this, we envision extending the workflow to support more generalized and intuitive input forms. For instance, users could provide a real-world video as input, from which the system could automatically extract relevant keyframes for subsequent coordination. Alternatively, generative model-based methods could be integrated to support user-driven creation directly from semantic-level inputs. On the output side, we also see potential for diversification. Our current implementation presents the output as a dynamic video clip. Depending on the context and narrative goals, this can be extended to support a wider range of storytelling scenarios like infographics or data news.

Supporting a more customized and mixed-initiative creation process. Our user study revealed significant diversity in user preferences related to visual design. Some users prioritized the clear presentation of data, favoring representations grounded in statistical accuracy, while others preferred more imaginative and expressive visual forms. For some, semantic coherence was critical, whereas others were more drawn to visual aesthetics and emotional appeal. Incorporating user preferences as conditioning factors within the model's reasoning and decision-making processes is crucial to accommodate this diversity. This enables a more personalized design experience while fostering effective human-AI collaboration [44,47,50]. Such a human-in-the-loop approach also helps navigate the trade-off between data fidelity and creative flexibility. Furthermore, introducing direct or hybrid interaction methods, such as selecting, linking, and direct manipulation of visual elements, can make the creation process more intuitive [63].

LLM performance for creative support. The flexibility of LLMs makes them well-suited for creative support tasks [46,48,52]. However, in our current workflow, we have observed several challenges that affect their reliability, efficiency, and creative diversity. Due to the complexity of task-specific reasoning and multimodal inputs, the response time of LLMs can vary. Each case takes on average 77.6 seconds, with 13,803 input and 1,832 output tokens required to generate a single result. While this cost is generally acceptable, handling multiple candidates introduces moderate additional overhead, highlighting the importance of efficient pruning and scheduling. Such latency can affect the system's ability to provide timely feedback, which is critical for maintaining user engagement and creative flow. Although presenting intermediate reasoning steps in the interface offers valuable insights, users still encounter inconsistent waiting periods. To improve responsiveness, we can explore strategies such as optimizing reasoning pipelines and adopting asynchronous response mechanisms [14]. Another challenge lies in the model's limited familiarity with tool usage. Despite efforts to standardize tool descriptions and prompt design, the model may still hallucinate unsupported functions in complex scenarios. This suggests the need to further expand the tool library [4] and explore fine-tuning approaches [12]. Additionally, while we incorporate design knowledge to support more grounded and comprehensive suggestions, the model still exhibits preferences. Understanding and mitigating such biases remains an important direction for future work.

Limitations and future work. Our evaluation strategy primarily relies on user studies to assess users' experiences with the system and their satisfaction with the resulting designs. Although SceneLoom received encouraging feedback in this limited-scale study, we recognize the need to broaden the participant base in future work. Involving a more diverse user base and collecting richer feedback enables iterative refinement of the system and its design space. Moreover, current forms of data visualization are mainly based on basic chart types and are refined through simple modifications, which limits their expressive power. In future work, we aim to explore more creative and visually compelling forms of data visualization. One promising direction involves leveraging generative models to synthesize entire visualizations and accompanying elements such as icons and illustrations. Furthermore, as the creative process becomes increasingly collaborative [32], we envision supporting co-creation and content sharing within the system. For instance, a community-oriented platform, like Pinterest, could be developed to facilitate the exchange of ideas, design inspiration, and peer feedback. Such a platform would further foster creativity through social interaction and collective refinement of visual concepts.

7 Conclusion

This paper presents SceneLoom, a VLM-powered system that enables the coordinated integration of data visualizations and real-world imagery for expressive data storytelling. Grounded in a formative study, we identified key design considerations. SceneLoom uses VLMs to extract and reason over these features, generating diverse, contextually aligned design alternatives that support narrative intent. Users can explore, refine, and animate these designs to externalize ideas and enhance visual communication. Our user study and example gallery validate SceneLoom's ability to inspire creativity and expand the expressive potential of narrative visualization. We hope our approach could help users enhance creative data communication and inspire future work.

Acknowledgments

This work is supported by Natural Science Foundation of China (NSFC No.62472099) and Ji Hua Laboratory S&T Program (X250881UG250). We sincerely thank Prof. Xingyu Lan for her valuable feedback.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] O. Avrahami, D. Lischinski, and O. Fried. Blended diffusion for text-driven editing of natural images. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2022, pp. 18187–18197, 2022.

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

[4] T. Cai, X. Wang, T. Ma, X. Chen, and D. Zhou. Large language models as tool makers. In *Proc. ICLR*, 2024.

[5] S. K. Card, J. Mackinlay, and B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[6] Q. Chen, W. Shuai, J. Zhang, Z. Sun, and N. Cao. Beyond numbers: Creating analogies to enhance data comprehension and communication with generative ai. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24. ACM, 2024.

[7] P. Cheng, L. Lin, J. Lyu, Y. Huang, W. Luo, and X. Tang. Prior: Prototype representation joint learning from medical images and reports. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, vol. 2023, pp. 21304–21314, 2023.

[8] L. B. Chilton, E. J. Ozmen, S. H. Ross, and V. Liu. Visifit: Structuring iterative improvement for novice designers. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, 2021.

[9] L. B. Chilton, S. Petridis, and M. Agrawala. Visiblends: A flexible workflow for visual blends. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, p. 1–14. ACM, 2019.

[10] D. Choi, S. Hong, J. Park, J. J. Y. Chung, and J. Kim. Creativeconnect: Supporting reference recombination for graphic design ideation with generative ai. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24. ACM, 2024.

[11] D. Coelho and K. Mueller. Infomages: Embedding Data into Thematic Images. *Computer Graphics Forum*, 2020.

[12] L. Gao, J. Lu, Z. Shao, Z. Lin, S. Yue, C. Leong, Y. Sun, R. J. Zauner, Z. Wei, and S. Chen. Fine-tuned large language model for visualization system: A study on self-regulated learning in education. *IEEE Trans. Vis. Comput. Graph.*, 31(1):514–524, 2025.

[13] J. Garnier. anime.js: Lightweight javascript animation library. https://github.com/juliangarnier/anime, 2019.

[14] A. A. Ginart, N. Kodali, J. Lee, C. Xiong, S. Savarese, and J. Emmons. Asynchronous tool usage for real-time agents. *arXiv preprint arXiv:2410.21620*, 2024.

[15] G. Gu, B. Ko, S. Go, S.-H. Lee, J. Lee, and M. Shin. Towards light-weight and real-time line segment detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1):726–734, 2022.

[16] T. Guardian. Edward snowden and the nsa files: facts and figures, 2014. https://www.youtube.com/watch?v=OFCNqkDWMtY&t=110s. Accessed: 2025-03-23.

[17] G. Guo, J. J. Kang, R. S. Shah, H. Pfister, and S. Varma. Understanding graphical perception in data visualization through zero-shot prompting of vision-language models. *arXiv preprint arXiv:2411.00257*, 2024.

[18] Q. Guo, S. De Mello, H. Yin, W. Byeon, K. C. Cheung, Y. Yu, P. Luo, and S. Liu. Regiongpt: Towards region understanding vision language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13796–13806, 2024.

[19] S. He and E. Adar. Vizitcards: A card-based toolkit for infovis design education. *IEEE Trans. Vis. Comput. Graph.*, 23(1):561–570, 2017.

[20] B. Herman, C. D. Jackson, and D. F. Keefe. Touching the ground: Evaluating the effectiveness of data physicalizations for spatial data analysis tasks. *IEEE Trans. Vis. Comput. Graph.*, 31(1):875–885, 2025.

[21] A. Hosseini, A. Kazerouni, S. Akhavan, M. Brudno, and B. Taati. Sum: Saliency unification through mamba for visual attention modeling. In *Proc. WACV*, pp. 1597–1607, 2025.

[22] D. Huang, M. Tory, B. Adriel Aseniero, L. Bartram, S. Bateman, S. Carpendale, A. Tang, and R. Woodbury. Personal visualization and personal visual analytics. *IEEE Trans. Vis. Comput. Graph.*, 21(3):420–433, 2015.

[23] M. S. Islam, R. Rahman, A. Masry, M. T. R. Laskar, M. T. Nayeem, and E. Hoque. Are large vision language models up to the challenge of chart comprehension and reasoning. In *Findings of EMNLP 2024*, pp. 3334–3368. ACL, 2024.

[24] T. W. S. Journal. This chinese restaurant chain built its $9b empire off customer service, 2024. https://www.youtube.com/watch?v=0jci98uOrWQ&t=170s. Accessed: 2025-03-23.

[25] Y. Kang, Z. Sun, S. Wang, Z. Huang, Z. Wu, and X. Ma. Metamap: Supporting visual metaphor ideation through multi-dimensional example-based exploration. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21. ACM, 2021.

[26] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, vol. 2023, pp. 3992–4003, 2023.

[27] W. Kong, Z. Jiang, S. Sun, Z. Guo, W. Cui, T. Liu, J. Lou, and D. Zhang. Aesthetics++: Refining graphic designs by exploring design principles and human preference. *IEEE Trans. Vis. Comput. Graph.*, 29(6):3093–3104, 2023.

[28] A. Kouts, L. Besançon, M. Sedlmair, and B. Lee. Lsdvis: Hallucinatory data visualisations in real world environments. *arXiv preprint arXiv:2312.11144*, 2023.

[29] J. Kuruvilla, D. Sukumaran, A. Sankar, and S. P. Joy. A review on image processing and image segmentation. In *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*, pp. 198–203, 2016.

[30] S. Laing and M. Masoodian. A study of the influence of visual imagery on graphic design ideation. *Design Studies*, 45:187–209, 2016.

[31] X. Lan, Y. Wu, and N. Cao. Affective visualization design: Leveraging the emotional impact of data. *IEEE Trans. Vis. Comput. Graph.*, 30(1):1–11, 2024.

[32] B. Lee, X. Hu, M. Cordeil, A. Prouzeau, B. Jenny, and T. Dwyer. Shared surfaces and spaces: Collaborative data visualisation in a co-located immersive environment. *IEEE Trans. Vis. Comput. Graph.*, 27(2):1171–1181, 2021.

[33] F. Li, H. Zhang, P. Sun, X. Zou, S. Liu, C. Li, J. Yang, L. Zhang, and J. Gao. Segment and recognize anything at any granularity. In *18th European Conference on Computer Vision, ECCV'24*, p. 467–484. Springer-Verlag, 2024.

[34] H. Li, L. Ying, L. Shen, Y. Wang, Y. Wu, and H. Qu. Composing Data Stories with Meta Relations. *arXiv preprint arXiv:2501.03603*, 2025.

[35] Z. Li, C. Gebhardt, Y. Inglin, N. Steck, P. Streli, and C. Holz. Situationadapt: Contextual ui optimization in mixed reality with situation awareness via llm reasoning. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24. ACM, 2024.

[36] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[37] Z. Liu, X. Xie, M. He, W. Zhao, Y. Wu, L. Cheng, H. Zhang, and Y. Wu. Smartboard: Visual exploration of team tactics with llm agent. *IEEE Trans. Vis. Comput. Graph.*, 31(1):23–33, 2025.

[38] M. Lukáč, J. Fišer, J.-C. Bazin, O. Jamriška, A. Sorkine-Hornung, and D. Sýkora. Painting by feature: texture boundaries for example-based image creation. *ACM Trans. Graph.*, 32(4), article no. 116, 2013.

[39] A. Lundgard and A. Satyanarayan. Accessible visualization via natural language descriptions: A four-level model of semantic content. *IEEE Trans. Vis. Comput. Graph.*, 28(1):1073–1083, 2022.

[40] M. MacKelvie. The clutch goat...(it's not who you think). https://www.youtube.com/watch?v=qjjW1l9KjXQ&t=763s. Accessed: 2025-03-23.

[41] F. Meng, J. Wang, C. Li, Q. Lu, H. Tian, T. Yang, J. Liao, X. Zhu, J. Dai, Y. Qiao, P. Luo, K. Zhang, and W. Shao. MMIU: Multimodal multi-image understanding for evaluating large vision-language models. In *Proc. ICLR*, 2025.

[42] L. Morais, Y. Jansen, N. Andrade, and P. Dragicevic. Showing data about people: A design space of anthropographics. *IEEE Trans. Vis. Comput. Graph.*, 28(3):1661–1679, 2022.

[43] OpenAI. tiktoken: A fast bpe tokeniser for use with openai's models. https://github.com/openai/tiktoken. Accessed: 2025-06-29.

[44] Y. Ouyang, L. Shen, Y. Wang, and Q. Li. NotePlayer: Engaging Computational Notebooks for Dynamic Presentation of Analytical Processes. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–20. ACM, 2024.

[45] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-Resolution Image Synthesis with Latent Diffusion Models . pp. 10674–

10685. IEEE, 2022.

[46] Z. Shao, L. Shen, H. Li, Y. Shan, H. Qu, Y. Wang, and S. Chen. Narrative Player: Reviving Data Narratives with Visuals. *IEEE Trans. Vis. Comput. Graph.*, pp. 1–15, 2025.

[47] L. Shen, H. Li, Y. Wang, T. Luo, Y. Luo, and H. Qu. Data playwright: Authoring data videos with annotated narration. *IEEE Trans. Vis. Comput. Graph.*, pp. 1–14, 2024.

[48] L. Shen, H. Li, Y. Wang, and H. Qu. From Data to Story: Towards Automatic Animated Data Video Creation with LLM-Based Multi-Agent Systems. In *IEEE VIS 2024 Workshop on Data Storytelling in an Era of Generative AI, GEN4DS'24*, pp. 20–27. IEEE, 2024.

[49] L. Shen, H. Li, Y. Wang, and H. Qu. Reflecting on Design Paradigms of Animated Data Video Tools. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1–21. ACM, 2025.

[50] L. Shen, H. Li, Y. Wang, X. Xie, and H. Qu. Prompting Generative AI with Interaction-Augmented Instructions. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, CHI EA '25*, pp. 1–9. ACM, 2025.

[51] L. Shen, E. Shen, Y. Luo, X. Yang, X. Hu, X. Zhang, Z. Tai, and J. Wang. Towards natural language interfaces for data visualization: A survey. *IEEE Trans. Vis. Comput. Graph.*, 29(6):3121–3144, 2023.

[52] L. Shen, Y. Zhang, H. Zhang, and Y. Wang. Data Player: Automatic Generation of Data Videos with Narration-Animation Interplay. *IEEE Trans. Vis. Comput. Graph.*, 30(1):109–119, 2024.

[53] L. Shen, Y. Zhang, H. Zhang, and Y. Wang. Data player: Automatic generation of data videos with narration-animation interplay. *IEEE Trans. Vis. Comput. Graph.*, 30(1):109–119, 2024.

[54] X. Shi, M. Liu, Z. Zhou, A. Neshati, R. Rossi, and J. Zhao. Exploring interactive color palettes for abstraction-driven exploratory image colorization. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24. ACM, 2024.

[55] X. Shi, Y. Wang, R. Rossi, and J. Zhao. Brickify: Enabling expressive design intent specification through direct manipulation on design tokens. *arXiv preprint arXiv:2502.21219*, 2025.

[56] Y. Shi, X. Lan, J. Li, Z. Li, and N. Cao. Communicating with motion: A design space for animated visual narratives in data videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21. ACM, 2021.

[57] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein. Saliency in vr: How do people explore virtual environments? *IEEE Trans. Vis. Comput. Graph.*, 24(4):1633–1642, 2018.

[58] T. Tang, J. Tang, J. Lai, L. Ying, Y. Wu, L. Yu, and P. Ren. Smartshots: An optimization approach for generating videos with data visualizations embedded. *ACM Trans. Interact. Intell. Syst.*, 12(1), 2022.

[59] W. Tong, K. Shigyo, L.-P. Yuan, M. Fan, T.-C. Pong, H. Qu, and M. Xia. Vistellar: Embedding data visualization to short-form videos using mobile augmented reality. *IEEE Trans. Vis. Comput. Graph.*, 31(3):1862–1874, 2025.

[60] Vox. World cup penalty kicks, tracked, 2023. https://www.youtube.com/watch?v=HAuwPue57Vs&t=151s. Accessed: 2025-03-23.

[61] F. Wang, B. Wang, X. Shu, Z. Liu, Z. Shao, C. Liu, and S. Chen. Chartinsighter: An approach for mitigating hallucination in time-series chart summary generation with a benchmark dataset. *arXiv preprint arXiv:2501.09349*, 2025.

[62] W. Wang, Z. Chen, X. Chen, J. Wu, X. Zhu, G. Zeng, P. Luo, T. Lu, J. Zhou, Y. Qiao, and J. Dai. Visionllm: large language model is also an open-ended decoder for vision-centric tasks. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, article no. 2688. Curran Associates Inc., 2023.

[63] Y. Wang, L. Shen, Z. You, X. Shu, B. Lee, J. Thompson, H. Zhang, and D. Zhang. WonderFlow: Narration-Centric Design of Animated Data Videos. *IEEE Trans. Vis. Comput. Graph.*, pp. 1–17, 2024.

[64] Z. Wang, A. Li, Z. Li, and X. Liu. Genartist: Multimodal LLM as an agent for unified image generation and editing. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[65] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22. Curran Associates Inc., 2022.

[66] W. Willett, Y. Jansen, and P. Dragicevic. Embedded data representations. *IEEE Trans. Vis. Comput. Graph.*, 23(1):461–470, 2017.

[67] J. Wu, J. J. Y. Chung, and E. Adar. viz2viz: Prompt-driven stylized visualization generation using a diffusion model. *arXiv preprint arXiv:2304.01919*, 2023.

[68] Y. Wu, L. Yan, L. Shen, Y. Wang, N. Tang, and Y. Luo. ChartInsights: Evaluating Multimodal Large Language Models for Low-Level Chart Question Answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 12174–12200. ACL, 2024.

[69] S. Xiao, S. Huang, Y. Lin, Y. Ye, and W. Zeng. Let the chart spark: Embedding semantic context into chart with text-to-image generative model. *IEEE Trans. Vis. Comput. Graph.*, 30(1):284–294, 2024.

[70] R. Xiaofeng and L. Bo. Discriminatively trained sparse code gradients for contour detection. In *Advances in Neural Information Processing Systems*, vol. 25. Curran Associates, Inc., 2012.

[71] Xinpianchang. What have we gone through to control the yellow river, 2024. https://www.xinpianchang.com/a12453678?kw=Guangxi%20Nationalities%20Museum. Accessed: 2025-03-23.

[72] L. Yang, X. Xu, X. Lan, Z. Liu, S. Guo, Y. Shi, H. Qu, and N. Cao. A design space for applying the freytag's pyramid structure to data stories. *IEEE Trans. Vis. Comput. Graph.*, 28(1):922–932, 2022.

[73] Z. Yang, L. Li, J. Wang, K. Lin, E. Azarnasab, F. Ahmed, Z. Liu, C. Liu, M. Zeng, and L. Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

[74] L. Yao, F. Bucchieri, V. McArthur, A. Bezerianos, and P. Isenberg. User experience of visualizations in motion: A case study and design considerations. *IEEE Trans. Vis. Comput. Graph.*, 31(1):174–184, 2025.

[75] L. Yao, R. Vuillemot, A. Bezerianos, and P. Isenberg. Designing for visualization in motion: Embedding visualizations in swimming videos. *IEEE Trans. Vis. Comput. Graph.*, 30(3):1821–1836, 2024.

[76] Y. Ye, R. Huang, and W. Zeng. Visatlas: An image-based exploration and query system for large visualization collections via neural image embedding. *IEEE Trans. Vis. Comput. Graph.*, 30(7):3224–3240, 2024.

[77] L. Ying, T. Tang, Y. Luo, L. Shen, X. Xie, L. Yu, and Y. Wu. Glyphcreator: Towards example-based automatic generation of circular glyphs. *IEEE Trans. Vis. Comput. Graph.*, 28(1):400–410, 2022.

[78] T. Yu, R. Feng, R. Feng, J. Liu, X. Jin, W. Zeng, and Z. Chen. Inpaint anything: Segment anything meets image inpainting. *arXiv preprint arXiv:2304.06790*, 2023.

[79] L.-P. Yuan, Z. Zhou, J. Zhao, Y. Guo, F. Du, and H. Qu. Infocolorizer: Interactive recommendation of color palettes for infographics. *IEEE Trans. Vis. Comput. Graph.*, 28(12):4252–4266, 2022.

[80] X. Zeng, H. Lin, Y. Ye, and W. Zeng. Advancing multimodal large language models in chart question answering with visualization-referenced instruction tuning. *IEEE Trans. Vis. Comput. Graph.*, 31(1):525–535, 2025.

[81] J. E. Zhang, N. Sultanum, A. Bezerianos, and F. Chevalier. Dataquilt: Extracting visual elements from images to craft pictorial visualizations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, p. 1–13. ACM, 2020.

[82] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, vol. 3836-3847, pp. 3813–3824, 2023.

[83] J. Zhou, R. Li, J. Tang, T. Tang, H. Li, W. Cui, and Y. Wu. Understanding nonlinear collaboration between human and ai agents: A co-design framework for creative design. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24. ACM, 2024.

[84] M. Zhou, D. Zhang, W. You, Z. Yu, Y. Wu, C. Pan, H. Liu, T. Lao, and P. Chen. Stylefactory: Towards better style alignment in image creation through style-strength-based control and evaluation. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24. ACM, 2024.

[85] T. Zhou, G. Y.-Y. Chan, S. Guo, J. Hoffswell, C. Xiao, V. S. Bursztyn, and E. Koh. Data pictorial: Deconstructing raster images for data-aware animated vector posters. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST'24. ACM, 2024.

[86] C. Zhu-Tian, Y. Wang, Q. Wang, Y. Wang, and H. Qu. Towards automated infographic design: Deep learning-based auto-extraction of extensible timeline. *IEEE Trans. Vis. Comput. Graph.*, 26(1):917–926, 2020.

[87] C. Zhu-Tian, Q. Yang, X. Xie, J. Beyer, H. Xia, Y. Wu, and H. Pfister. Sporthesia: Augmenting sports videos using natural language. *IEEE Trans. Vis. Comput. Graph.*, 29(1):918–928, 2023.

[88] C. Zhu-Tian, S. Ye, X. Chu, H. Xia, H. Zhang, H. Qu, and Y. Wu. Augmenting sports videos with viscommentator. *IEEE Trans. Vis. Comput. Graph.*, 28(1):824–834, 2022.

Supplementary Materials

A Implementation Details

This section presents the prompt examples described in Sec.4, with their corresponding output examples.

A.1 Data Preparation

This section provides examples of prompts and output results mentioned in Sec.4.2, including feature extraction from narrative intent and data table and data visualization generation.

Prompt template for feature extraction:
You are an expert in data analysis and data-driven storytelling. You need to extract data-related information, entity objects, and actions to guide animations based on the narrative intent {*narrative intent*} and the corresponding data table {*data*}. Some explanations are as follows:

Data-related information

- Data fact: Extract all key data points and insights directly from the narration, avoiding redundancy. Insights should be derived primarily from the narration, with data used as supporting evidence rather than being the sole source of information.

- Metadata: Provide structured metadata to support visualization and data transformation, including relevant data columns and data transformation methods.

Entity objects: Identify real-world objects or concepts that have explicit semantic meaning and could be represented visually.

Actions

- Enter: Describe how elements should appear in the visualization, including the sequence or transition style.

- Emphasize: Highlight key information using animation techniques, specifying what should be emphasized and how.

The output should be in the following JSON format:

```
{
  "dataRelatedInformation": {
    "dataFact": [],
    "metadata": {
      "dataColumns": [],
      "dataTransformation": []
    }
  },
  "entityObjects": [],
  "actions": [
    {"enter": ""},
    {"emphasize": ""},
  ]
}
```

An example of extracted features:

```
{
  "dataRelatedInformation": {
    "dataFact": [
      "46% of Americans prefer artificial trees, while 26% prefer real ones.",
      "Over recent decades, there has been a steady increase in the use of artificial trees."
    ],
    "metadata": {
      "dataColumns": ["Response", "Share of Respondents (%)"],
      "dataTransformation": [
        "Sort 'Response' column in descending order based on 'Share of Respondents (%)'."
      ]
    }
  },
  "entityObjects": ["Artificial Christmas tree", "Real Christmas tree", "Survey respondents"],
  "actions": [
    {
      "enter": "Introduce the growing preference for artificial trees over real trees using a transition animation."
    },
    {
      "emphasize": "Highlight the 46% artificial tree preference in the US by making the corresponding bar pulse or grow slightly."
    }
  ]
}
```

Data visualizations are generated using d3.js. We have designed multiple d3.js-based visualization interfaces to create consistent yet flexible visualization variants (e.g., horizontal histogram, vertical histogram). Thus, the task here is to generate the required interface parameters. Furthermore, we leverage data-related information from prior feature extraction to produce more related visualization alternatives.

Prompt template for data visualization generation:
You are an expert in data visualization and data analysis. You need to generate as many appropriate visualization charts as possible based on the data information {*dataRelatedInformation*} and data table {*data*}.

The output should be in the following JSON format:

```
{
  //"bar-0", "line-0", "line-1"
  "chartId": "",
  "type": "",
  //Categorical data, only list the column name
  "categoryKey": [],
  //Numerical data, only list the column name(s)
  "valueKeys": [],
  //Title of the chart
  "title": ""
}
```

An example of visualization generation parameters:

```
[
    {
        "chartId": "bar-0",
        "type": "vertical",
        "categoryKey": ["Response"],
        "valueKeys": ["Share of Respondents (%)"],
        "title": "Distribution of Respondents",
    },
    {
        "chartId": "bar-1",
        "type": "horizontal",
        "categoryKey": ["Response"],
        "valueKeys": ["Share of Respondents (%)"],
        "title": "Distribution of Respondents",
    },
    {
        "chartId": "line-0",
        "type": "basic",
        "categoryKey": ["Response"],
        "valueKeys": ["Share of Respondents (%)"],
        "title": "Trends in Tree Preferences",
    },
    {
        "chartId": "line-1",
        "type": "with-dot",
        "categoryKey": ["Response"],
        "valueKeys": ["Share of Respondents (%)"],
        "title": "Trends in Tree Preferences",
    },
    {
        "chartId": "line-2",
        "type": "with-area",
        "categoryKey": ["Response"],
```

```
        "valueKeys": ["Share of Respondents (%)"],
        "title": "Trends in Tree Preferences",
    },
    {
        "chartId": "scatter-0",
        "type": "basic",
        "categoryKey": ["Response"],
        "valueKeys": ["Share of Respondents (%)"],
        "title": "Respondent Share by Response Type",
    },
    {
        "chartId": "scatter-1",
        "type": "with-size",
        "categoryKey": ["Response"],
        "valueKeys": ["Share of Respondents (%)"],
        "title": "Respondent Share by Response Type",
    },
    {
        "chartId": "area-0",
        "type": "basic",
        "categoryKey": ["Response"],
        "valueKeys": ["Share of Respondents (%)"],
        "title": "Proportion of Tree Choices",
    }
]
```

A.2 Visual Perception

In this section, we present the generated specifications from visualization interpretation and real-world element understanding. Additionally, the image information is provided in the format: {"type": "image_url", "image_url": {}}.

Prompt template for visualization specification:
The user will give you a visualization of the data in SVG form {*SVG*} and the corresponding visualization in PNG form. Please combine these two outputs with the specification in JSON format:

```
class VisDescription {
/** Chart type (e.g., bar, line, point, area) */
  chartType: string;
  /** Spatial layout and coordinate axes */
  spatialSubstrate: {
    /** Axis definitions mapping data fields */
    axis: { x: string; y: string };
    /** Chart layout variant */
    chartVariants: string;
  }
  /** Visual elements in the chart */
  graphicalElements: {
    mark: {
      /** Mark type (e.g., rect, line, point, arc) */
      type: string;
      role: "dataMarker" | "annotation";
      /** Graphical properties and encoding methods */
      graphicalProperties: {
        [key: string]: string;
      }
    }[]
  }
  /** Main insight derived from the visualization */
  visualInsight: string;
}
```

An example of bar chart specification:

```
{
    "visDescription": {
        "chartType": "bar",
        "spatialSubstrate": {
            "axis": {"x": "Continent", "y": "Value" },
            "chartVariants": "stacked"
        },
        "graphicalElements": {
            "mark": [
                {
                    "type": "bar",
                    "role": "dataMarker",
                    "graphicalProperties": {
                        "height": "The height of each bar
                            represents the magnitude of
                            tree cover change for each
                            continent. Bars extend both
                            upward and downward from the x-
                            axis, showing the gross gain
                            and gross loss respectively.
                            Taller bars indicate greater
                            amounts of change.",
                        "color": "Gross gain is represented
                             by light brown (beige) bars,
                            while gross loss is shown in
                            light blue bars. Both colors
                            distinguish the two components
                             of tree cover change in a
                            stacked bar layout."
                    }
                }
            ]
        },
        "visualInsight": "The visualization shows the
            gross gain and loss of tree cover across
            various continents. Global has the largest
            total with both high gross gain and loss.
            Other continents vary, with differing
            patterns in their gains and losses."
    }
}
```

Regarding the description of elements in the real-world image, we also provide entity object information extracted from the narrative intent to assist VLMs in achieving a more accurate understanding of semantic information.

Prompt template for real-world element specification:
The user will provide you with two images: the original image and the masked image of the object in the image outlined with a blue bounding box. The user will also provide background information about the image, which includes a narration describing the related event and extracted semantic objects to assist in generating semantic content. Please combine these two images to output a JSON data format description of the masked object:

```
/** Grain level description: element grouping and
     geometric type */
class grainLevel {
  /** Single element or grouped elements */
  type: 'singleElement' | 'groupedElements'
  /** Geometric primitive(s): point, line, or plane */
  geometricPrimitive: string | string[]
}

/** Element description within a single element */
class elementDescription {
  /** Spatial layout description of the element */
  layout: string
  /** Geometric shape of the element (e.g., circle,
       rectangle) */
  shape: string
  /** Semantic role or meaning of the element */
  semantic: string
}

/** Element level description, grouped by geometric
     primitives */
class ElementLevel {
```

```
  /** Plane-level elements (each described by layout,
       shape, and semantic role) */
  plane: ElementDescription[]
}

class imageDescription {
  /** Grain level: grouping and geometry type */
  grainLevel: GrainLevel
  /** Element level: specific element descriptions */
  elementLevel: ElementLevel
}
```

An example of the single element specification:

```
{
    "imageDescription": {
        "grainLevel": {
            "type": "singleElement",
            "shapeType": "line"
        },
        "elementLevel": {
            "line": {
                "layoutDescription": "A horizontal
                     structure spans across a river or
                     small body of water between trees.",
                "shape": "flattening",
                "semantic": "bridge"
            }
        }
    }
}
```

An example of grouped elements specification:

```
{
    "imageDescription": {
        "grainLevel": {
            "type": "groupedElements",
            "shapeTypes": [
                "plane"
            ],
            "distributionLayout": "linear"
        },
        "elementLevel": {
            "plane": [
                {
                    "layoutDescription": "The object
                         extends vertically along the
                         center left of the image.",
                    "shape": "rectangle",
                    "semantic": "Real Christmas tree"
                },
                {
                    "layoutDescription": "The object
                         extends vertically along the
                         center right of the image.",
                    "shape": "rectangle",
                    "semantic": "Artificial Christmas tree
                         "
                }
            ]
        }
    }
}
```

A.3 Reasoning and Mapping

In this section, we present the design and generation of the mapping, along with the associated reasoning process. This includes adjustments to data visualization, as well as the invocation of tools and configuration of parameters during the implementation stage.

The input of design mapping generation consists of four components:

- *Real-world scene image:* A photograph where specific elements are segmented and outlined with blue contours.
- `imageDescription` *(JSON file):* A structured file that describes each segmented element in the form of specification defined before.
- `visDescription` *(JSON file):* A specification of the data visualization as mentioned before.
- `visSVG` *(SVG file):* A vector-based representation of the visualization, which contains all visual elements with unique class names.

Prompt template for design mapping:
You are a data analyst and designer specializing in integrating data visualizations into real-world scene images based on narrative intent. Your task is to analyze the visual features and semantic structures of these inputs and generate creative design proposals that seamlessly integrate the real-world scene with the data visualization.

Based on prior design experiences, you should refer to the following design patterns to guide your proposal. Both semantic coherence and visual alignment are important. While achieving both is ideal, satisfying one dimension can still produce effective results.

1. Spatial Organization.
A single element in the real-world scene can be mapped to a data marker or a set of data markers sharing the same data attributes, the entire canvas, or coordinate axes. When `type` is `singleElement` under `grainLevel` in `imageDescription`. Grouped elements can be mapped to data markers but require a data-binding relationship. When `type` is `groupedElements` under `grainLevel` in `imageDescription`.

2. Shape Similarity.
It involves two types: similar in shape types and similar in shape features. For shape types, the shapes of real-world elements approximate the mark types in data visualizations, such as points to points, lines to lines, and circles to circles. Refer to the `shapeType` under `grainLevel` in `imageDescription` and the `chartType` or `type` under `mark` in `visDescription`. For shape features, the visual shape features of real-world elements can correspond to the mark `type` or `chartType` in the visualization or point to insights from the overall visual representation `visualInsight` of the data visualization.

3. Layout Consistency
We consider relative positions and the distribution here to meet the layout alignment. The relative position of individual elements within a real-world scene can correspond to the spatial layout of a visualization, like serving as the coordinate origin. Consider the `elementLevel` under `layoutDescription` in the `imageDescription` and the `spatialSubstrate` in the `visDescription`. The distribution of grouped elements within a real-world scene corresponds to the overall visualization placement. This should be considered regarding the `distributionLayout` under `grainLevel` in `imageDescription` and the `spatialSubstrate` or `visualInsight` in `visDescription`.

4. Semantic Binding
The semantics of real-world entities can directly correspond to the meaning of the data or metaphorically represent data categories. Additionally, elements conveying narrative context can also be mapped accordingly. This rule can be considered by referring to the `semantic` in `imageDescription` and metadata information in `visualInsight`.

If no design mapping exists, return None. If a design mapping exists, please provide your design proposal in the following JSON structure:

```
/** Overview of the entire design plan */
class designPlan {
  /** General overview description of the design plan
```

```
      */
  overview: string
  /** List of mapping plans connecting real-world
      elements to visualization elements */
  mappingPlan: MappingPlan[]
}

/** Description of a single mapping between real-world
      and visualization elements */
class mappingPlan {
  /** Type of mapping */
  mappingType: 'one-to-one' | 'one-to-many' | 'many-to-
      many'
  /** Names or IDs of real-world elements involved */
  realWorldElements: string[]
  /** Names or IDs of visualization elements involved
      */
  visElements: string[]
  /** Optional: semantic alignment if available */
  semanticAlignment?: SemanticAlignment
  /** Optional: visual alignment if available */
  visualAlignment?: VisualAlignment
}

/** Description of semantic alignment between data
     visualization and real-world elements */
class semanticAlignment {
  /** Semantic meaning in the data visualization (e.g.,
       category, metric, label, auxiliary fuction) */
  visSemantic: string
  /** Semantic meaning in the real-world element (e.g.,
       object role, contextual meaning) */
  realWorldSemantic: string
  /** Explanation of the semantic relationship and its
      intended effect */
  description: string
}


/** Description of visual alignment: shape and layout
     matching */
class visualAlignment {
  /** Optional: shape alignment if available */
  shapeAlignment?: ShapeAlignment
  /** Optional: layout alignment if available */
  layoutAlignment?: LayoutAlignment
}

/** Shape alignment: visual shape mapping description
     */
class shapeAlignment {
  /** Shape or visual feature of the real-world element
       */
  realWorldShape: string
  /** Corresponding visualization element shape */
  visShape: string
  /** Explanation of shape alignment logic and its
      visual effect */
  description: string
}

/** Layout alignment: spatial arrangement mapping
     description */
class layoutAlignment {
  /** Layout or positioning of real-world elements */
  realWorldLayout: string
  /** Layout or positioning in the visualization */
  visLayout: string
  /** Alignment type: e.g., center, bottom-left, bottom
      -right, top-left, top-right */
  alignmentType: 'center' | 'bottom-left' | 'bottom-
      right' | 'top-left' | 'top-right'
  /** Explanation of layout alignment and its spatial
      effect */
  description: string
}
```

An example of mapping design:

```
{
  "designPlan": {
    "overview": "Integrate the Ferris wheel in the real-
        world scene with the donut chart visual,
        aligning the circular shapes for thematic
        cohesion and enhancing storytelling through
        visual symbolism.",
    "mappingPlan": [
      {
        "realWorldElements": ["#ferris-wheel"],
        "mappingType": "one-to-one",
        "visElements": ["#donut-chart"],
        "semanticAlignment": {
          "dataSemantic": "Age groups in merchandise
              sales",
          "realWorldSemantic": "Ferris wheel symbolizing
              cycles and diversity",
          "description": "The Ferris wheel metaphorically
               represents the cyclical and inclusive
              nature of age diversity in merchandise
              sales."
        },
        "visualAlignment": {
          "shapeAlignment": {
            "realWorldShape": "Circle",
            "visShape": "Donut",
            "description": "Both the Ferris wheel and the
                donut chart share a circular shape."
          },
          "layoutAlignment": {
            "realWorldLayout": "Upper-left quadrant of
               the scene",
            "visLayout": "Center of the visualization
               canvas",
            "alignmentType": "center",
            "description": "The prominent position of the
                Ferris wheel in the upper-left quadrant
                aligns with the central placement of
               the donut chart."
          }
        }
      }
    ]
  }
}
```

For all design mappings, we submit a request and recommendation regarding the necessity of data visualization adjustments, which include both data-level and view-level modifications. If the model determines that appropriate visualization adjustments can achieve improved mapping without altering the correctness of the data fact, it can utilize relevant parameters and return a corresponding list of functions.

Prompt template for visualization adjustment and tool execution:

You may propose reasonable modifications to the visualization view to support the integration, but these changes must respect visualization principles and maintain data narrative consistency. Modifications can occur both at the data level and the view level.

At the data level, you are allowed to improve entity-to-data mapping by filtering or pruning the dataset using the function `filterData(dataset, filterCondition)`, where `dataset` is the input data collection and `filterCondition` defines the rule for selecting relevant entities. You can also sort the dataset to highlight key insights by applying `sortData(dataset, sortKey, sortOrder)`, where `sortKey` specifies the attribute for sorting and `sortOrder` determines whether the sorting is ascending, descending or other orders. To better structure the information, you may categorize the data using `categorizeData(dataset, categoryKey)`, grouping entities based on a shared attribute.

At the view level, adjustments should aim to better align visual elements with real-world scene representations while preserving clarity and meaning. You may resize elements by applying `editSvgSize(SvgElement, targetHeight, targetWidth)`, shift their positions using `editSvgPosition(SvgElement, targetX, targetY)`, or adjust their orientation through `editSvgRotation(SvgElement, targetAngle)`. Before that, you should select and align the anchor point to modify these operations using `alignToElement(source, target, alignmentType)`.

Finally, you should return a sequence of the applied operations in the form of function calls with their arguments.

An example of function calls is as follows:

```
[
  "alignToElement('radarChart', 'cityCenter', 'center')",
  "editSvgPosition('radarChart', cityCenter.center.x, cityCenter.center.y)",
  "editSvgSize('radarChart', cityCenter.boundingBox.xmax - cityCenter.boundingBox.xmin, cityCenter.boundingBox.ymax - cityCenter.boundingBox.ymin)"
]
```

All elements specified in the mapping plan are represented using class names for element access. We employ the following approach for parameter storage and access for the basic parameters of point, line, and plane elements in real-world scenes.

```
/** Point element: describes key attributes for alignment */
class Point {
  /** x coordinate of the point */
  x: number
  /** y coordinate of the point */
  y: number
  /** Size of the point */
  size?: number
  /** Color of the point */
  color?: string
  /** Semantic label */
  label?: string
  /** Bounding box of the point */
  boundingBox?: { xmin: number, ymin: number, xmax: number, ymax: number }
}

/** Line element: describes key attributes for alignment */
class Line {
  /** Start x coordinate */
  x1: number
  /** Start y coordinate */
  y1: number
  /** End x coordinate */
  x2: number
  /** End y coordinate */
  y2: number
  /** Width of the line */
  width?: number
  /** Color of the line */
  color?: string
  /** Semantic label */
  label?: string
  /** Center point of the line (optional, pre-calculated) */
  center?: { sx: number, y: number}
  /** Length of the line */
  length?: number
  /** Angle of the line to the horizontal */
  angle?: number
  /** Bounding box of the line */
  boundingBox?: { xmin: number, ymin: number, xmax: number, ymax: number }
}

/** Plane element: describes key attributes for alignment */
class Plane {
  /** List of boundary points defining the plane */
  boundaryPoints: { x: number, y: number}[]
  /** Semantic label */
  label?: string
  /** Center point of the plane */
  center: {x: number, y: number}
  /** Shape type: e.g., rectangle, polygon, circle */
  shapeType?: string
  /** Aspect ratio (width / height) of the plane */
  aspectRatio?: number
  /** Bounding box of the plane */
  boundingBox?: { xmin: number, ymin: number, xmax: number, ymax: number }
}
```

The relevant tools and parameters are listed in Table. 1.

Table 1: Tools for manipulating SVG elements in data visualization.

Tool	Parameters
getSvgElement	(SvgPath, className)
editSvgSize	(SvgElement, targetHeight, targetWidth)
editSvgPosition	(SvgElement, targetX, targetY)
editSvgRotation	(SvgElement, targetAngle)
alignToElement	(source, target, alignmentType)

A.4 Design Evaluation

This section presents our method for evaluating and providing suggestions for a design alternative, including assessments of data accuracy and visual clarity. Particular attention is given to determining whether instances of data conflict require handling through inpainting operations.

Prompt template for design evaluation:
We are evaluating the effectiveness of a visualization that combines data graphics with real-world imagery. A structured data table will be provided as the ground truth. You are asked to assess the visualization based on both the visual content and the provided data table.

For data accuracy, you should evaluate whether the integration of visual elements (charts, marks, overlays) with the image accurately conveys the underlying data, and whether data values, trends, or relationships are clearly and correctly represented. Additionally, you must check for any conflicts between the visualization and the data table. If a conflict is detected, determine whether inpainting is necessary. If inpainting is needed, you should provide the coordinates of the point where correction should occur (point_coords), and assess whether there are existing elements that can be reused (reusable_element). If a reusable element is available, the method remove_anything.py should be applied. If no reusable element is available, the method fill_anything.py should be used instead, and a semantic text prompt (text_prompt) must be provided to guide the inpainting process.

For readability and clarity, you should assess whether the visualization is easy to understand at a glance, whether the incorporation of the image enhances or hinders the viewer's interpretation of the data, and whether visual elements such as labels, highlights, colors, and scales are clear and distinguishable.

Your evaluation should include a score for each category on a scale of 1 (very poor) to 5 (excellent), accompanied by a brief explanation supporting your assessment. The data table should be used to substantiate your evaluation of accuracy.

Please format your evaluation results in the following JSON structure:

```
/** Evaluation result for design effectiveness */
class evaluationResult {
  /** Evaluation of data representation accuracy */
  data_accuracy: DataAccuracy
  /** Evaluation of visualization readability and
       clarity */
  readability: Readability
}

/** Details about data accuracy evaluation */
class dataAccuracy {
  /** Score from 1 (very poor) to 5 (excellent) */
  score: number
  /** Brief explanation supporting the score */
  explanation: string
  /** Whether a conflict between visualization and
       ground-truth data is detected */
  conflict_detected: boolean
  /** Inpainting operation details if a conflict exists
        */
  inpaint_operation?: InpaintOperation
}

/** Inpainting operation specification */
class inpaintOperation {
  /** Whether inpainting is required */
  need_inpaint: boolean
  /** Coordinates indicating correction points */
  point_coords: {
    /** X coordinate of the correction point */
    x: number
    /** Y coordinate of the correction point */
    y: number
  }[]
  /** Whether an existing visual element can be reused
       */
  reusable_element: boolean
  /** Selected method for inpainting: reuse or semantic
        fill */
  method: 'remove_anything.py' | 'fill_anything.py'
  /** Textual prompt describing the semantic content if
        using "fill_anything.py" */
  text_prompt?: string
}

/** Details about readability evaluation */
class readability {
  /** Score from 1 (very poor) to 5 (excellent) */
  score: number
  /** Brief explanation supporting the score */
  explanation: string
}
```

An example of design evaluation:

```
{
  "data_accuracy": {
    "score": 3,
    "explanation": "Data points incorrectly placed.",
    "conflict_detected": true,
    "inpaint_operation": {
      "need_inpaint": true,
      "point_coords": [
        {"x": 320,"y": 210},
        {"x": 450,"y": 310}
      ],
      "reusable_element": true,
      "method": "fill_anything.py",
      "text_prompt": "the blue sky."
    }
  },
  "readability": {
    "score": 3,
    "explanation": "Visualization is mostly clear,
         slight label overlaps."
  }
}
```

The command templates for executing removal or filling operations via inpainting are as follows:

```
python remove_anything.py \
    --input_img {input_img_path} \
    --coords_type {coords_type} \
    --point_coords {point_coords} \
    --point_labels {point_labels} \
    --dilate_kernel_size {dilate_kernel_size} \
    --output_dir {output_dir} \
    --sam_model_type {sam_model_type} \
    --sam_ckpt {sam_ckpt_path} \
    --lama_config {lama_config_path} \
    --lama_ckpt {lama_ckpt_path}

python fill_anything.py \
    --input_img {input_img_path} \
    --coords_type {coords_type} \
    --point_coords {point_coords} \
    --point_labels {point_labels} \
    --text_prompt "{text_prompt}" \
    --dilate_kernel_size {dilate_kernel_size} \
    --output_dir {output_dir} \
    --sam_model_type {sam_model_type} \
    --sam_ckpt {sam_ckpt_path}
```

A.5 Animation Generation

This section demonstrates the prompt for generating animations for SVG-based design alternatives. We will use the action descriptions obtained from feature extraction in narrative intents as references to generate Anime.js [13] code for animation production. The prompt design refers to the related work by Shen et al. [47, 53] to guide the model in understanding the relationships between data visualization and animation types.

Prompt template for animation generation:

You are an expert in generating animations for SVG elements using anime.js. Based on the natural language description of the animation {*actions*} and the provided SVG file {*SVG*}, first identify the target elements that require animation. Then, according to the description and the characteristics of data visualization, generate the corresponding anime.js animation code.

You may use the following animation guidelines as a reference:

- Axes-fade-in: Apply changes in opacity and strokeWidth to elements with the class '.axis' (opacity: [0,1], strokeWidth: [0,2], duration: 800) — used for rendering coordinate axes.
- Bar-grow-in: Animate height and translateY with elasticity (height: ['0%', '100%'], translateY: [50,0], elasticity: 300, stagger: 100) — used for animating bar chart columns.
- Line-wipe-in: Animate strokeDashoffset (strokeDashoffset: [anime.setDashoffset, 0], duration: 1500) — used for line chart paths.
- Pie-wheel-in: Apply rotation and scaling (rotate: ['-90deg', '0deg'], scale: [0,1], easing: 'spring(1, 80, 10, 0)') — used for animating pie chart sectors.
- Float-in: Animate translateY/translateX and opacity (translateY: ['20px', '0'], opacity: [0,1], direction: 'top', delay: 200) — used for

floating tooltips or annotations.
- Change-color: Change the fill color (fill: ['ccc', 'f00'], duration: 500) for elements with the selector *e.g.,* '.bar[data-value>50]'.

Please return the animation in the following JSON structure:

```
/** Configuration for anime.js animation */
class animeJSConfig {
  /** CSS selector, DOM element, or array of elements
       to animate */
  targets: string | HTMLElement | HTMLElement[]
  /** Type of animation: entrance, emphasis, or exit */
  animation_type: 'entrance' | 'emphasis'
  /** Animation properties depending on the animation
       type */
  properties: {
    /** Key-value pairs for property animation, e.g.,
         opacity: [0, 1] */
    [propertyName: string]: [string | number, string |
         number]
    /** Direction of animation movement */
    direction?: 'top' | 'bottom' | 'left' | 'right'
  }
  /** Timing control for the animation */
  timing: {
    /** Total duration of the animation in milliseconds
          */
    duration: number
    /** Delay before the animation starts in
         milliseconds */
    delay: number
  }
}
```

An example of generated animation for bar chart:

```
{
  targets: '.bar',
  animation_type: 'entrance',
  properties: {
    scaleY: [0,1],
    opacity: [0,1],
    direction: 'bottom'
  },
  timing: {
    duration: 800,
    delay: 100
  }
}
```

B Evaluation Details

We provide supplementary materials to support our evaluation of SceneLoom, including: (1) a small-scale assessment of LLM usage cost during the generation process, (2) a detailed explanation of the user study procedure, and (3) the complete questionnaires and response results for the user study. In addition, design outcomes produced by participants are presented in a packaged format within the *Examples* folder.

B.1 LLM Usage Cost

To assess the efficiency and resource consumption of the generation process, we track two key metrics throughout the workflow: *Time Elapsed* and *Token Usage*. We measure the total time required for LLM to generate the output using Python's time module. Timestamps are recorded before and after the model call, and the difference yields the elapsed time in seconds. Token usage is estimated using OpenAI's tiktoken library [43] to encode the input tokens and the model's response. The total token count is computed by summing the number of tokens in each of these segments.

We further evaluated a set of five representative cases across a three-stage workflow: data preparation, which involves generating the initial visualizations, extracting key features, and constructing structured specifications; reasoning and mapping, where mapping strategies are generated and relevant tools are invoked and adjusted accordingly; and evaluation, which assesses the accuracy of data representations, visual clarity, and attention guidance. In each case, we also recorded the number of generated design elements or alternatives, as this directly affects both computation time and potential economic cost.

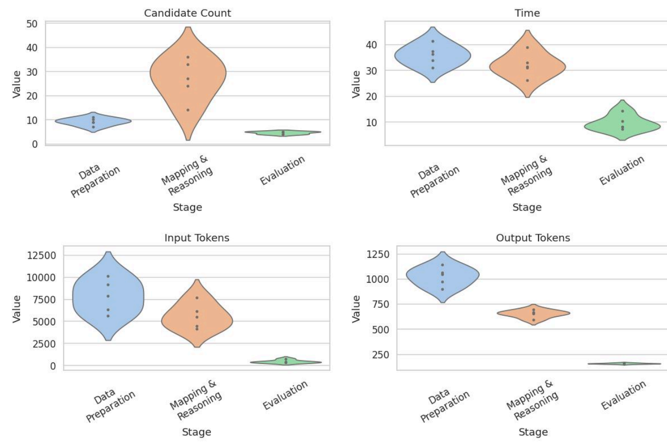

Fig. 10: Distribution of four computational metrics.

To understand the end-to-end computational cost, we aggregate metric values across the three workflow stages, as illustrated in Fig. 10. On average, each case takes 77.6 seconds, consumes 13,803 input tokens, and produces 1,832 output tokens. The largest variance is observed in input token usage, indicating differences in input length and reasoning complexity.

Across each stage, we observe a consistent trend in time and token usage. Mapping & Reasoning is the most resource-intensive stage, with the highest candidate count (M=26.8), while Evaluation is the most lightweight, both in execution time (M=9.2) and token consumption (M=433.0; M=157.3). These results highlight the middle stage as the key computational bottleneck and suggest potential for optimization through candidate pruning or reasoning efficiency.

B.2 User Study Protocol

Participants: 10 individuals were invited to participate in the evaluation study, with an even distribution of 4 male and 6 female participants.

Procedure:

1. The purpose and process of our user study will be explained to all participants. We will also present five representative examples from our collected corpus to more vividly illustrate the research content and experimental objectives to the participants. (duration: 5min)

2. Participants are first required to complete a ***self-report*** evaluating their level of expertise or experience in fields related to data visualization, visual design, and video editing. We employ a 5-point Likert scale, where participants rate their expertise, with 1 indicating no experience and 5 indicating expert-level proficiency. (duration: 5min)

3. We present ten groups of datasets covering different themes and formats. After a brief introduction, participants independently select two datasets as tasks in the subsequent experiment. We guide participants to gain a deeper understanding of each dataset and encourage them to freely articulate their ideas on how to approach the design, either through verbal descriptions or by sketching. (duration: 10min)

4. We first introduce the basic workflow and interaction features of our system through an example. Participants then use SceneLoom to work on the two assigned tasks. During the creation process, we require them to think aloud while we observe their design activities. Finally, we save the participants' final design outcomes. (duration: 20min)

5. At the end of the study, we ask participants to complete a ***subjective questionnaire*** and a semi-structured interview regarding their usage experience and satisfaction with SceneLoom. The questions involved in the questionnaire are shown in Table. 2. (duration: 10min)

Table 2: Domains and associated questions in user evaluation

Domain	Question
Usability	Q1: Satisfied with interface design and interactions. Q2: Easy to understand and use.
Effectiveness	Q3: Enables exploratory and creative design process. Q4: Satisfied with the design outcomes.
Recommendations	Q5: Engaging and enjoyable design experience. Q6: Likely to recommend SceneLoom to others.

B.3 Questionnaires and Results

The self-report scores of the 10 participants (P1-P10) regarding their level of expertise on data visualization, visual design and video editing are presented in Table. 3. Among them, P1 and P2 are data analysts, P3 is a journalist, P4 and P8 are HCI researchers, P5 and P7 are designers, and P6 and P9 are VIS researchers.

Table 3: Participant self-reports on experience levels

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	$M \pm Std$
Vis	4	3	3	4	3	5	5	4	5	2	3.8 ± 1.03
Design	2	3	3	3	5	4	4	4	3	3	3.4 ± 0.84
Video	4	2	5	2	3	2	4	5	4	3	3.4 ± 1.17

User feedback on their experience with SceneLoom and their satisfaction with the design outcomes is presented in Table. 4. The results in this table are consistent with the user rating results for the subjective questions as presented in the main text.

Table 4: Participant ratings on user experience of SceneLoom

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	$M \pm Std$
Q1	4	4	5	4	5	5	4	4	5	4	4.4 ± 0.52
Q2	5	5	5	4	4	5	4	3	5	4	4.4 ± 0.70
Q3	4	4	5	5	5	5	4	5	5	5	4.7 ± 0.48
Q4	4	5	4	4	4	5	4	5	2	3	4.0 ± 0.94
Q5	4	4	5	5	5	5	5	5	4	3	4.5 ± 0.71
Q6	4	5	4	5	5	5	4	5	5	4	4.6 ± 0.52