Sahit Kadthala

Professor Leo Zhicheng Liu

May 13, 2025

Infographic Generator: LLM-Driven Visualization of Narrative Content

1. Introduction

Creating expressive infographics from narrative text remains a complex challenge in visualization authoring. While many existing tools focus on data-driven storytelling or manually structured input, less attention has been paid to automating the process for unstructured, prose-style narratives. Such narratives are common in domains like education and journalism, where the author wishes to convey a sequence of events or processes in a visually digestible format.

In this paper, we introduce a novel application that leverages the power of large language models to bridge this gap. Our system allows users to input natural language stories and automatically generates infographic-style outputs by decomposing the text into semantically distinct stages, identifying key visual entities, suggesting representative icons, and initializing a default canvas layout. Unlike prior tools that rely on structured data or predefined message templates, our application emphasizes open-ended storytelling and uses an LLM to interpret, visualize, and scaffold the authoring process.

We contribute a pipeline that combines automated narrative parsing with interactive visual refinement. By integrating an LLM as both a content extractor and a design assistant, our approach supports users through both the idea generation and design phases of visual storytelling.

2. Related Work

Our work builds on prior research in visualization authoring tools, particularly systems that facilitate the transformation of narrative content into visual representations. Several tools have explored the space of authoring expressive data-driven or story-driven visuals through both manual and semi-automated means.

Epigraphics introduces a mixed-initiative system for generating infographics from messages structured by the user [1]. The tool presents a message-driven pipeline that allows users to iteratively create visual infographic designs based on what they are trying to communicate. While our system shares the goal of streamlining infographic generation, our approach focuses on unstructured narrative inputs rather than short, structured message inputs. We extend the idea of message extraction by leveraging large language models (LLMs) to automatically identify key stakeholders and objects from natural language stories, and then assign them to narrative stages for visualization. Our use of an LLM generalizes beyond the constraints of predefined message structures, allowing for more flexible authoring that emphasizes the user's narrative.

DataClips presents a system for authoring short data-driven videos by pairing narrative text with visuals like charts and images [3]. It emphasizes the manual composition of scenes aligned with structured data insights. In contrast, our system focuses on unstructured narratives, using LLMs to automatically segment stories, extract visualizable concepts, and suggest default icon placements. This automation expands the storytelling space to more expressive, non-data-centric narratives, reducing the authoring burden for users.

Timeline Storyteller shows how interaction design can be leveraged by users to construct more expressive, timeline-based visualizations that better represent time-based narratives [4]. Although our system does not focus on temporal data, we adopt a similar philosophy in offering a structured layout, in the form of stage canvases, that support the placement and refinement of visual elements across frames. Like Timeline Storyteller, we aim to lower the barrier for users to produce rich, customized storytelling outputs.

Studies have shown that investing in the ideation process during early stages of design can help users articulate and refine their goals through writing, which in turn clarifies the intended message of the visualization [2]. Our system incorporates this idea by encouraging users to begin with a story, from which the system extracts structure and meaning, guiding the creation of the final infographic through informed defaults and editable suggestions. We integrated LLMs not only as a content extractor but also as a creative collaborator that suggests relevant visual mappings upfront, providing users with an immediate starting point for further authoring.

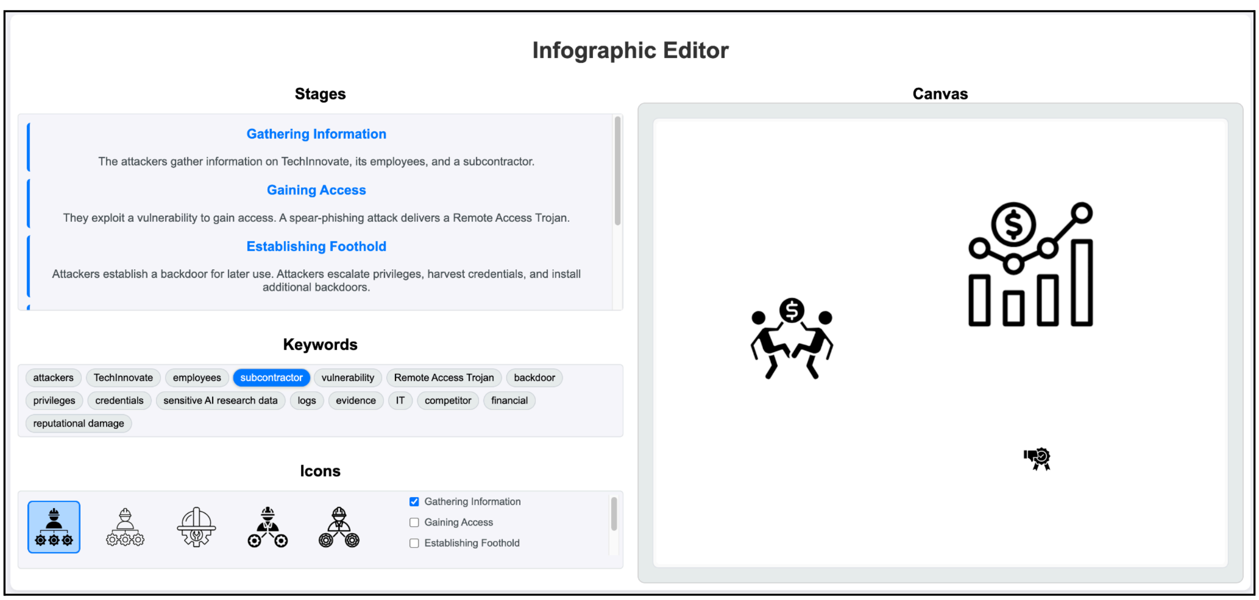
Recent work has also explored the use of AI and language interfaces in authoring data videos and visual explanations. Data Video is a tool that allows users to create animated explanatory videos from natural language descriptions by linking them with appropriate data facts and transitions [5]. While their focus is on structured data and animations, we adapt a similar ideology of leveraging natural language descriptions to lower authoring complexity, so that users can create infographic videos faster.

Research has been done in the area of using declarative natural language instructions to generate explanatory visualizations [6]. It has been determined that there is value in using natural language as a programming interface for storytelling, a principle that supports our approach. However, we target a more free-flowing narrative use case, rather than analytical explanations. Broader work shows that tools like ours are within a growing ecosystem of narrative-based visualization systems [7]. The main design goals of such systems are storytelling expressiveness, interpretability, and customizability, which reinforces our decision to prioritize human-in-the-loop control and provide options to edit the AI-driven suggestions in our tool.

These previous systems underscore the growing interest in authoring tools that blend automation with human guidance. Our application contributes to this space by offering a novel pipeline that takes natural language as input and combines narrative decomposition, visual entity identification, and layout suggestion. We build on prior advances while expanding toward more open-ended use cases that require unstructured narrative inputs.

3. Approach

Our system is designed to support the authoring of infographic-style narratives from unstructured text. It draws on insights from visualization authoring tools and narrative design research, while introducing LLM-driven automation to scaffold the storytelling and visual encoding process. The system follows a multi-stage pipeline, blending natural language understanding with user-driven refinement. This section outlines the high-level methodology across four core components.



3.1 Story Interpretation and Stage Segmentation

The process begins when the user inputs a freeform narrative, often in the form of paragraphs, describing a sequence of events, process, or situation. Instead of requiring structured data or formal annotations, we leverage an LLM to divide the story into coherent stages. Each stage represents a semantically distinct frame of the narrative and forms the backbone of the visual structure. We prompt the LLM to return a CSV of stage titles and their associated sentence-level content, preserving the original language of the story while organizing it into interpretable segments. This framing mirrors the structure of data-driven storytelling techniques seen in prior systems, such as DataClips and Timeline Storyteller, but introduces a lightweight, language-driven mechanism for segmentation that doesn't depend on inputs following a specific data format.

3.2 Entity Extraction and Keyword Identification

Once the narrative is segmented, the next step involves extracting keywords, specifically, entities and concepts that are meaningful within the context of the story and may be important to be visualized in the infographic. This would include stakeholders (e.g., organizations, people, attackers) and objects (e.g., emails, devices, and vehicles) that play a role across one or more of the narrative stages. This design captures what Epigraphics refers to as "message-driven visuals," which are visuals grounded in salient narrative concepts rather than purely quantitative contexts. By identifying and extracting keywords in natural language input, we open up the possibility of generating visual representations that are more semantically relevant and communicative.

3.3 Icon Recommendation and Stage Association

For each extracted keyword, the system retrieves at most the top five relevant SVG thumbnails to serve as options for the icon that will represent the keyword in the infographic canvas. The top suggested icon is automatically the default representation of the keyword, enabling the canvas to populate meaningfully without requiring immediate user intervention. We also leverage the LLM to map each of the keywords to however many stages they appear in, using natural language context to infer co-occurrence. This results in a preliminary assignment of icons to stages. Users retain the ability to manually revise icon selections and check/uncheck associated stages in the interface, but the system initializes these values based on semantic defaults.

3.4 Layout Suggestion and Canvas Initialization

Finally, the system proposes a default layout for icons on the canvas. We prompt the LLM and ask it to assign default locations on the canvas. Then, the user has the ability to drag and resize icons freely into their final positions. Having a default layout encourages expressiveness without overwhelming the user with having to create an infographic from a blank canvas. Our approach positions LLMs as layout assistants rather than final arbiters, generating default placements that users can refine into the final infographic.

4. Technical Details

This section outlines the implementation of our application, focusing on the architecture, data flow, LLM prompting strategies, and client-side interaction mechanisms. The system is built using standard web technologies with integration of LLM APIs and external icon search services.

4.1 System Architecture

The application consists of three main components: a frontend UI, an LLM integration layer, and the icon retrieval API. The frontend UI is built with HTML, CSS, and JavaScript, supporting user input, stage navigation, icon selection, and canvas manipulation. As for the LLM integration, the LLM used in this project is known as 'deepseek-r1-distill-llama-70b' and is a free-to-use model that can be accessed via the OpenRouter interface. The LLM is tasked with handling text segmentation, keyword extraction, stage-to-keyword mapping, and layout suggestion. The icon retrieval API is from The Noun Project and provides the ability to fetch relevant icons for each keyword. It uses OAuth-based access and returns a filtered set of image thumbnails. All generated outputs, including the stages, keywords, mappings, and layouts, are stored in sessionStorage, allowing the results page (results.html) to reconstruct the entire visual state without requiring backend persistence.

4.2 Story Segmentation

When the user submits a story, two back-to-back LLM calls are made. The first call is the stage segmentation prompt, which leads the LLM to return a CSV mapping stage names to their respective content from the story, ensuring that the content in the stages remains the exact words from the original narrative. The second call is the keyword extraction prompt, which results in a comma-separated list of icon-friendly stakeholders and objects that can be searched for in the Noun Project API using GET. Both calls are prompt-engineered in a way such that the output is always structured and predictable. Any identified edge cases in the output have been handled with error-handling code that cleans up the output into the desired results.

4.3 Stage Assignment and Icon Initialization

A third LLM call associates each keyword with the stage(s) in which it appears and is contextually relevant. This mapping is critical for both the canvas layout and the checkbox display logic. The mapping is returned as a CSV that maps each keyword to a list of its corresponding stage names. Upon keyword processing, the app fetches icons via the search GET request, provided by The Noun Project, and assigns the first icon as the default visual. All stages in the map are populated with their respective icons, ensuring that even if users do not interact with the keyword buttons, clicking a stage immediately reveals all assigned icons.

4.4 Canvas Behavior and Icon Layout

The canvas has several features that provide the user with additional precision in modifying the default infographic to communicate the narrative better. The first feature is Drag & Drop, which uses mousedown, mousemove, and mouseup event listeners to move icons around the canvas just by dragging them to their new position. The Resizing feature allows the user to change the size of each icon by clicking on a small square that appears in the bottom left when the user hovers over an icon. The code uses clamping logic to ensure that icons cannot be moved or resized outside the bounds of the canvas. As for the default layout suggestion, this mechanism is handled by a final LLM call that returns a CSV providing X and Y coordinates for each icon to fit into the canvas.

5. Case Study

To evaluate the application's ability to translate narrative text into meaningful visual representations, we conducted a case study using a security incident involving a fictional company, TechInnovate. The narrative details a multi-stage cyberattack, from reconnaissance to data exfiltration and its aftermath. The goal of this evaluation is to assess how effectively the system can identify and extract distinct stages and entities, and generate a coherent, icon-driven infographic.

The input story describes how attackers gather intelligence on TechInnovate, breach the system via spear-phishing, escalate privileges, and establish persistence mechanisms. Eventually, sensitive AI research is exploited, and the damage surfaces when IT launches an investigation. The breach culminates in financial and reputational harm caused by a competitor leveraging the stolen data.

5.1 LLM Prompting

We crafted targeted prompts that guide the language model through several stages of interpretation and transformation. For simplicity, the prompts refer to the input story as {story}. The first prompt instructs the model to segment the user-provided TechInnovate narrative into distinct stages.

```
`Parse this following story: {story}. I want you to divide up this story into a set of stages, where each stage will correspond to a different frame of an infographic video. The stages should be distinct enough to warrant a separate frame, and should be ordered in a logical sequence. The contents of each stage will be the exact sentences word-for-word from the source text. Name each of these stages and place only the name of the stage (not the stage number) before its corresponding sentences from the source text. Return a CSV of these stage names and contents. Make sure that both the stage names and contents are always enclosed in quotation marks. The first row should be the header row that has the names of the two columns, "Stage Name" and "Stage Content". The second row should be the first stage name and its corresponding content. The third row should be the second stage name and its corresponding content. And so on. Do not add anything extra like labeling different stages as stage 1, stage 2, etc. Simply return a CSV of the stages.`
```

Once stages are established, a second prompt is issued to extract the visualizable keywords from the full narrative, looking specifically for stakeholders, like TechInnovate, and objects, like Remote Access Trojan, that could be represented as icons.

```
`Parse this following story: {story}. I am trying to create an infographic about a given scenario provided below. I want you to read through the scenario and pick out the most important stakeholders that can be visualized using SVG icons in an infographic. Then, I want you to find the most important objects that can be visualized using SVG icons. For example, if there is an object “Computer” this should be picked as it is an important object that can be visualized easily with an icon. I want the format of the output to be a comma separated list of all the words you pick out. Do not add anything extra like labeling different words as stakeholders or objects. Simply return a comma separated list.`
```

A third prompt maps these keywords back to the stage structure by identifying all stages where a keyword is semantically relevant.

```
`Parse this following story: {story}. Parse this set of stages that corresponds directly to the story: {stages}. I am trying to create an infographic about the story and I split the story up into these different stages. For each of these keywords, ${keywords}, I want you to determine all of the stages that each of the keywords is
```

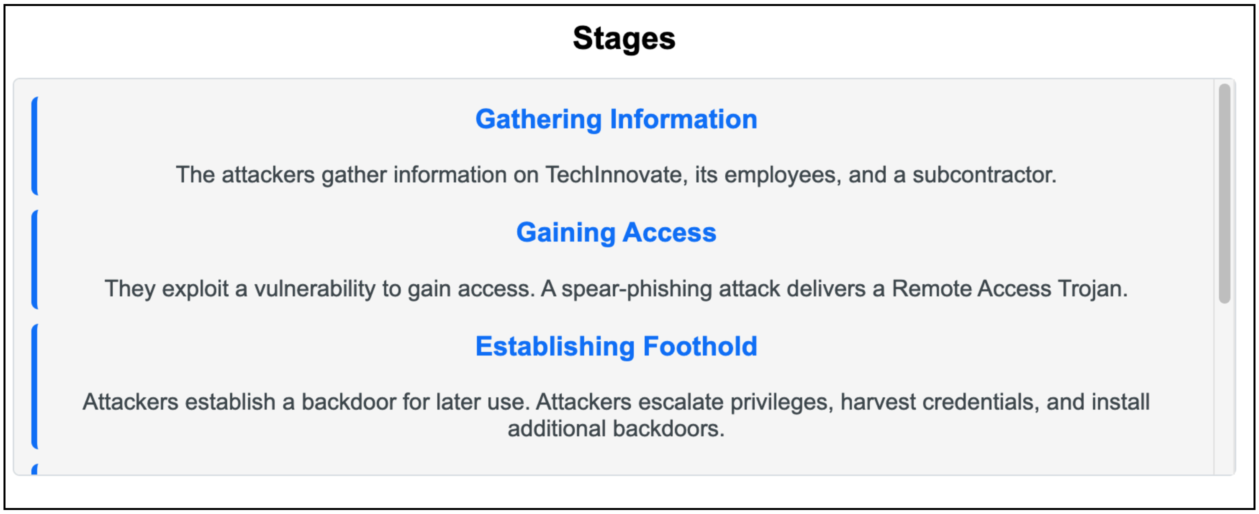
```
relevant in. Return a CSV of these keywords and stage names. Make sure that both the
keywords and stage names are always enclosed in quotation marks. The first row should
be the header row that has the names of the two columns, "Keywords" and "Stage Names".
The second row should be the first keyword and its corresponding stage names. The
third row should be the second keyword and its corresponding stage names. And so on.
Do not add anything extra like labeling different stages as stage 1, stage 2, etc.
Simply return a CSV of the keywords and their corresponding stage names. Do not
explain what the output is either, literally just give me an output of the CSV.`
```

This sequence of prompts enables the system to automatically suggest a visual representation of the TechInnovate narrative.

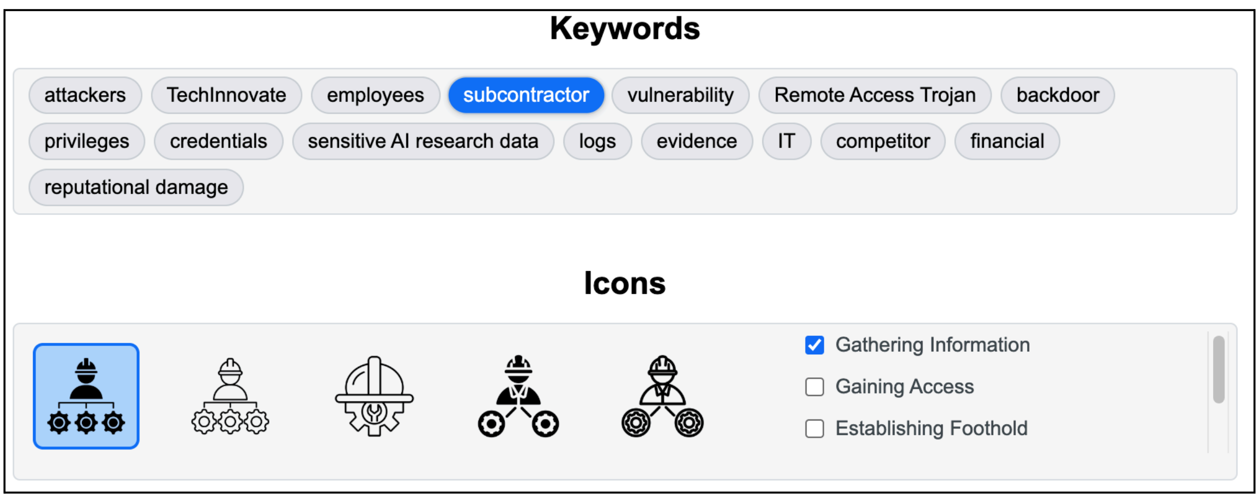
5.2 Output Breakdown

The application automatically segmented the narrative into coherent stages, which are shown in the ‘Stages’ section of the application. Each stage is followed by the exact text from the narrative that corresponds to it, so that users can see how the text that they provided was segmented into distinct sections. The names of the stages and short descriptions are as follows:

- Gathering Information: Describes the initial intelligence collection on employees and subcontractors.
- Gaining Access: Covers the exploit and delivery of the Remote Access Trojan.
- Establishing Foothold: Includes privilege escalation, harvesting of credentials, and the installation of persistent backdoors.
- Data Exfiltration: Focuses on the gradual theft of sensitive AI Research data.
- Discovery and Aftermath: Details the IT team’s discovery of anomalies, its investigation into the breach, and the resulting consequences.



Each of these stages was assigned relevant keywords extracted by the LLM, such as attackers, Remote Access Trojan, logs, and evidence. For each keyword, the system fetched a set of relevant icons and assigned the top result as the keyword's default image. These icons were automatically distributed across the appropriate stages in the canvas view.



The resulting infographic shows clear, non-overlapping visuals for each stage, which users can further refine through drag-and-drop and resizing interactions. This demonstrates how narrative content is transformed into an infographic with expressive visual frames.

6. Lessons Learned

Through the development of this system, several key lessons emerged about the collaborative relationship between humans and AI in the context of narrative infographic visualization. First, we found that using an LLM is highly effective at initiating the visual authoring process by generating stage breakdowns, extracting visualizable keywords, and proposing layout decisions. This significantly reduces the burden of starting from scratch because the user has initial results for all of the different parts of the infographic. The user can focus more on making sure that the final infographic is communicating their intended message.

However, we observed that LLM outputs require human interpretation and refinement to fully align with users' goals. The model is capable of identifying relevant concepts but lacks the sensitivity to distinguish between subtle narrative nuances. Some keywords may be too generic or too specific to represent visually without clarification. One example from the case study is the keyword 'logs', which is too generic and results in images of tree logs which do not fit the context of cybersecurity breaches. An example of a keyword that is too specific is 'TechInnovate' which is the name of a fictitious company, so no icon results are returned. This requires the system to search for an alternate term that would best represent this keyword in the given context, while ensuring that this alternate term is visualizable and available in the database.

We also learned the importance of prompt engineering and refining prompts to ensure that humans are getting exactly what they need from the AI. The final prompts that we use for this application are the product of countless iterations that ensure that we have the exact results needed for the background processes to run smoothly. For example, the result we asked for initially in all of our prompts was a CSV, but then we realized we had to provide examples of a CSV table with a couple of columns that we needed the LLM to fill in, in order to get our desired results. Without this specificity, the LLM had occasional formatting inconsistencies and occasional hallucinations that would cause the system to crash. By refining the prompt through extensive prompt engineering, as well as implementing troubleshooting safeguards, we ensured that the user gets the exact results that they need from the AI.

These observations underscore the evolving role of AI not as a replacement for human creativity, but as a collaborator that can accelerate and scaffold the design process. We found that the most effective results emerged when humans treated LLM outputs as suggestions rather than final answers. Rather than aiming for a fully autonomous generation, the design of our system benefits from co-creation between humans and AI, where the AI does the legwork of structure and inspiration, and the human provides judgment, allowing humans to remain in control of meaning.

7. Conclusion

This work presents a new approach to infographic generation that leverages the interpretive capabilities of LLMs to transform unstructured narratives into structured visual stories. Through a series of prompt-based steps, our system segments stories, extracts visualizable entities, assigns icons to narrative stages, and recommends default layouts, all without requiring structured data input. The result is a novel tool that lowers the barrier for creating expressive visuals from prose. Our evaluation demonstrates that this method can effectively support case-based storytelling in domains such as cybersecurity, where incident narratives are rich in structure but often hard to visualize. By combining intelligent defaults with manual editing capabilities, the system enables users to author infographics quickly while retaining creative control.

Our prototype does not yet constitute a complete infographic authoring system. Several key capabilities remain unimplemented, including cross-frame synchronization of recurring icons, intelligent layout algorithms to avoid icon overlap, and support for animations or transitions between frames. These are important directions for future work. We also envision extending the system to support multimodal inputs such as images or audio narration, and further tightening the semantic alignment between the input and the LLM results. We will continue to reimagine infographic creation as a co-creative process between humans and AI.

8. References

[1] Sashank Varma, Chengzhi Li, Bardia Doosti, Zhuoran Zheng, and Fanny Chevalier. 2024. *Epigraphics: Message-Driven Infographics Authoring*. Association for Computing Machinery, New York, NY, USA. https://dl.acm.org/doi/pdf/10.1145/3613904.3642172

[2] Ryan Kelly, Fanny Chevalier, and Petra Isenberg. 2024. *"It's a Good Idea to Put It Into Words": Writing 'Rudders' in the Initial Stages of Visualization Design*. https://arxiv.org/pdf/2407.15959

[3] Fereshteh Amini, Nathalie Henry Riche, Bongshin Lee, Andres Monroy-Hernandez, and Pourang Irani. 2016. *Authoring Data-Driven Videos with DataClips*. Microsoft Research Technical Report. https://fereshtehamini.github.io/assets/dataclips.pdf

[4] Bongshin Lee, Nathalie Henry Riche, Petra Isenberg, and Sheelagh Carpendale. 2018. *Timeline Storyteller: The Design & Deployment of an Interactive Authoring Tool for Expressive Timeline Narratives*. Microsoft Research. https://www.microsoft.com/en-us/research/uploads/prod/2018/12/TSCJ2019.pdf

[5] Mengwen Shi, Xiaoyang Tao, Fan Du, Zheguang Zhao, and Donghao Ren. 2021. *Data Video: An Authoring Tool for Data-Driven Videos Using Natural Language*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3411764.3445694

[6] Leixian Shen, Mengwen Shi, Fan Du, and Donghao Ren. 2023. *Instructions: Authoring Visual Explanations with Declarative Language*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3544548.3580752

[7] Leixian Shen, Mengwen Shi, Fan Du, and Donghao Ren. 2023. *Data Video Authoring: A Survey and Future Directions*. Computer Graphics Forum 42, 6 (2023), 1–21. https://doi.org/10.1111/cgf.14857