DOI: 10.1111/cgf.14324
Eurographics Conference on Visualization (EuroVis) 2021
R. Borgo, G. E. Marai, and T. von Landesberger
(Guest Editors)

Volume 40 (2021), Number 3

AutoClips: An Automatic Approach to Video Generation from Data Facts

D. Shi[1], F. Sun[1], X. Xu[1], X. Lan[1], D. Gotz[2], and N. Cao[1]

[1] Tongji University, Shanghai, China
[2] University of North Carolina at Chapel Hill, North Carolina, United States

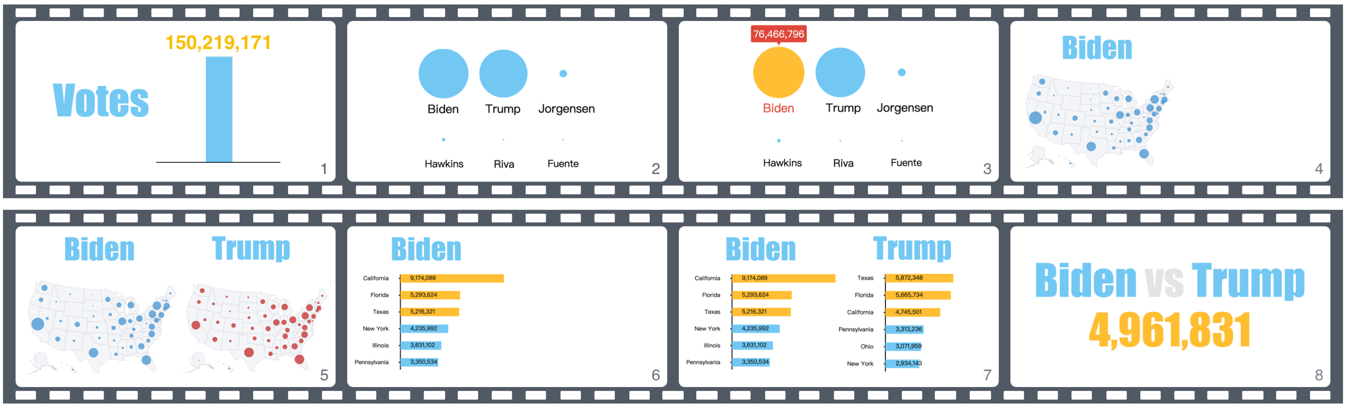

Figure 1: *This figure shows the key frames of a data video automatically generated by AutoClips from a sequence of data facts (see Table 1 in Section 5). The video tells a story about the 2020 US presidential election (open this link to view the video online).*

Abstract

Data videos, a storytelling genre that visualizes data facts with motion graphics, are gaining increasing popularity among data journalists, non-profits, and marketers to communicate data to broad audiences. However, crafting a data video is often time-consuming and asks for various domain knowledge such as data visualization, animation design, and screenwriting. Existing authoring tools usually enable users to edit and compose a set of templates manually, which still cost a lot of human effort. To further lower the barrier of creating data videos, this work introduces a new approach, AutoClips, which can automatically generate data videos given the input of a sequence of data facts. We built AutoClips through two stages. First, we constructed a fact-driven clip library where we mapped ten data facts to potential animated visualizations respectively by analyzing 230 online data videos and conducting interviews. Next, we constructed an algorithm that generates data videos from data facts through three steps: selecting and identifying the optimal clip for each of the data facts, arranging the clips into a coherent video, and optimizing the duration of the video. The results from two user studies indicated that the data videos generated by AutoClips are comprehensible, engaging, and have comparable quality with human-made videos.

CCS Concepts

• ***Human-centered computing*** → *Information visualization; Visualization toolkits; Visualization systems and tools;*

1. Introduction

Data videos, as a visual storytelling form that combines data visualization with motion graphics [AHRL*15, ARL*16], are among the seven narrative visualization genres proposed by Segel and Heer [SH10]. Given their ability to deliver condensed information efficiently and engage users with animation [ARL*18], data videos have been applied as an effective medium for data communication. For example, *Wealth Inequality in America* [Pol12], a data video about the polarization between the rich and the poor in the US, has gained more than 23 million views on video platforms and triggered heated discussion on social media [Jef13].

However, creating a data video is not easy; users not only need to craft animation for visualizations, but also have to synthesize the animated visualizations into a coherent story. Currently, few tools can achieve these two goals simultaneously in an efficient way. For example, programming tools such as D3 [BOH11], gganimate [PR], Gemini [KH20], and Canis [GZL*20] can generate animation for visualizations but are unable to compose stories. Design software such as Adobe After Effects [Ado] is good at editing motion but not specialized in creating data visualizations. In particular, mastering these tools requires expertise in either programming or design skills, which can be challenging for non-professional users. To ease the creation of data videos, some tools have been developed to support video authoring. For example, DataClips [ARL*16] provides a set of animation clips (e.g., creation, cycling, destruction) for visualizations such as bar charts, pictographs, and maps so that users can choose the clips they want and organize the chosen clips into a video. Similarly, Flourish [Ltd] developed various templates for animated visualizations (e.g., bar chart race); users can create a video by editing templates and arranging them into a sequence. Nevertheless, although these tools have improved the efficiency of creating data videos, they still require the users themselves to decide *what* animation should be applied to visualizations, and *how* to arrange the animated visualizations into a story. Making such decisions is still laborious and not easy, especially for novices.

Given these two challenges, we propose AutoClips, an automatic approach that generates data videos directly from a sequence of data facts (i.e., the atomic components of data stories such as *value*, *difference*, *proportion*, and *trend*, as defined by Wang *et al.* [WSZ*19]). First, we manually collected and analyzed 230 data videos in the wild to associate ten common data facts with potential video clips (a clip is a visualization animated with a series of motions to convey a data fact); we also built a fact-driven clip library based on the analysis. Then, we constructed an algorithm to generate data videos automatically. The algorithm is executed through three steps. First, given the input of a sequence of data facts, the algorithm selects clips from the fact-driven clip library and decides an optimal clip for each of the data facts. Second, the algorithm composes the clips together and optimizes the coherence of the video. Last, the algorithm configures the duration of the video to make it more compacted. To evaluate the effectiveness of AutoClips, we conducted two user studies with 30 and 24 participants, respectively. The first user study showed that the clips generated by AutoClips can present data facts appropriately. The second user study showed that the data videos generated by AutoClips are comparable to the videos manually composed by human designers. To sum up, the major contributions of this paper include:

- **A Library.** We constructed a fact-driven clip library by analyzing 230 data videos and conducting interviews with 12 users to associate ten data facts with typical animated visualizations.
- **An Algorithm.** We constructed an algorithm to generate data videos automatically from a sequence of data facts. The algorithm can select optimal clips to present data facts, arrange the clips into a coherent story, and optimize the duration of the story.
- **Evaluation.** We conducted two user studies to assess the effectiveness of AutoClips. The studies showed that the clips and data videos automatically generated by AutoClips are comprehensible and engaging. Also, the data videos produced by AutoClips are comparable to manually-composed data videos.

2. Related Work

This section overviews prior studies on data video, animation in visualization, and automatic data story generation.

2.1. Data Video

Data videos combine animation with data visualization to support data presentation and communication [AHRL*15]. As identified by Segel and Heer [SH10], data videos are among the seven main genres of narrative visualization. Compared to other genres, data videos are totally author-driven [SH10] so that viewers can follow the narratives in a linear manner. Also, data videos are good at engaging viewers since animated visualizations are attention-grabbing [ARL*18]. To leverage the benefits that data videos can bring, researchers have been studying how to design expressive data videos as well as how to create data videos more efficiently. For example, Amini *et al.* [AHRL*15] decomposed 50 data videos into establisher, initial, peak, and release, inspired by the concepts in cinematography. They then proposed a set of narrative structure patterns (e.g., establisher + initial + peak) in data videos. Recently, Tang *et al.* [TYT*20] proposed a taxonomy of narrative transition of data videos, identifying five types of typical transitions, of which Preserving Guide and Narrative Agent could preserve the content during the transition period.

Amini *et al.* [ARL*16] introduced an authoring tool called DataClips to facilitate the creation of data videos. They first analyzed more than 70 data videos to identify common building blocks (i.e., clips) in data videos. The clips are characterized by two dimensions, including visualization type (e.g., bar chart, line chart) and animation task (e.g., creation, destruction). Then, they developed a system that enables users to select clips and assemble the clips into a data video. Focused on time-series data, Lu *et al.* [LWY*20] proposed a video authoring tool that automatically extracted and presented changes of a time-series data. Similarly, tools such as Flourish [Ltd] in the wild also aid the creation of data videos by allowing users to edit and compose video templates.

While the aforementioned tools have eased the design process, creating a data video is still not easy, since users have to decide what visualizations and animations should be used and how to compose a coherent video by themselves. To save such effort, this work first introduces an automatic approach, AutoClips, to generating a data video automatically from a sequence of data facts.

2.2. Animation in Visualization

Animation is the change of visual representations over time [RFF*08]. In the visualization community, animation has been viewed as an important method to present data changes and facilitate the understanding of visualization transitions [Fis10, ARL*18]. For example, Heer and Robertson [HR07] found that appropriately-designed animation can significantly enhance the perception of visualization transitions. Thompson *et al.* [TLLS20] constructed the design space of animated data graphics, by identifying primitives in object, graphics, data, and timing dimensions, and compositing high-level transition types and pacing techniques with these primitives. Some researchers conducted experiments to evaluate how animation helps perform data analysis tasks [EDF08, APP10, BLIC19]. Meanwhile, many tools have been developed to support creating animated visualizations. For example, D3 [BOH11] provides a novel domain-specific JavaScript language to create animation for data visualizations flexibly. DynaVis [HR07] presents a C# language visualization framework to support creating animated transitions for statistical charts. Canis [GZL*20] is a high-level grammar for declaring the animation of charts in SVG. Gemini [KH20] performs and recommends animated transitions between two single-view Vega-Lite charts. Compared with these recent works on chart animation, AutoClips focuses on automatic video generation from data facts instead of specifying the animations' configuration in the video.

While the above studies focus on using animation to aid data analysis and exploration, how to generate animation for storytelling remains underexplored [CRP*16]. According to prior research, animation is a widely-observed method to link story chapters and guide story progression [SH10, AHRL*15]. Animation can also make visualizations more attractive and thus engage viewers [ARL*18]. However, few tools can support the automatic generation of animation for storytelling purpose now. Given such motivations, this work proposes a novel approach to generating animation for data videos automatically.

2.3. Automatic Data Story Generation

Recent years have witnessed an increasing popularity in generating data stories automatically, which asks for synthesizing individual visualizations into a certain story genre. For example, Wang *et al.* [WSZ*19] proposed DataShot, which can generate fact sheets from data facts automatically. In this work, they identified 11 categories of data facts (e.g., value, proportion, trend) by analyzing 245 fact sheets. Shi *et al.* [SXS*20] extended the idea of data facts and developed Calliope, a system that can automatically generate visual data stories in which the data facts are arranged into a sequence logically. In line with DataShot and Calliope, we use data facts as the input of our automatic algorithm. However, different from previous research, this work extends the automation of data stories to data videos, thus contributing new knowledge around automatic visualization design to the community.

3. AutoClips Overview

This section first introduces a motivation scenario for AutoClips. Then, we formulate the pipeline of AutoClips and present a set of high-level design requirements to guide our design.

3.1. Motivation Scenario

To motivate our work, we describe an example based on our first-hand experience cooperating with a data journalism team to run a data video channel on social medial platforms. Two of the authors were involved in the design team and participated in the creation of data videos. We found that the process of creating a data video usually comprises three main phases: First, the team collected a series of data facts around a certain topic, and constructed a storyline by assembling the data facts into a sequence. Then, the team chose data visualizations for the data facts and decided how to animate the visualizations by drawing a storyboard. Finally, the team realized the storyboard using design software such as Adobe After Effects in which they edited and combined the animated visualization until making a coherent data video. The second and third phase were most time-consuming. To select appropriate visualizations and animations for data facts, designers usually needed to discuss for several hours and then sketched a storyboard. To realize the storyboard, the designers had to edit the visualizations and animations frame by frame. Sometimes, the edit of animation would cost several days. Moreover, it was also common that designers had to transfer from one tool to another when creating data videos since currently few tools can enable creating data visualizations, animating visualizations, and composing a story at the same time. Given such a heavy workload, the team felt it challenging to produce data videos efficiently. They also acknowledged that this problem is critical because social media platforms usually seek for fast update to attract traffic. Thus, designers in the data video team sought handy tools to produce data videos efficiently and automatically.

3.2. AutoClips Pipeline

Based on the aforementioned findings from real-world practice, we decided that our work should focus on reducing the human effort made to select animated visualizations and assemble individual clips (i.e., the second and third phase of creating data videos in the motivation scenario). Considering tabular data is the most frequently used type in the daily scenario, our work focus on the videos based on this kind of data type. We formulate the pipeline of AutoClips in Fig 2. Below we explain the pipeline step by step.

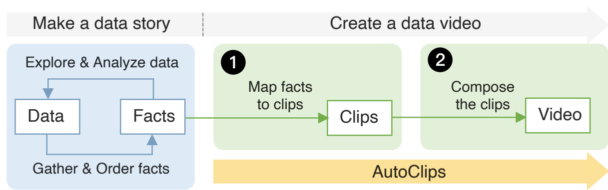

Figure 2: *The pipeline of AutoClips.*

Inputting Data Facts. The input of AutoClips is a sequence of data facts. According to the first definition in prior research [SDES18], a data fact describes the statistical data information used to create a visualization. Shi *et al.* [SXS*20] formulated a data fact into a 5-tuple structure, including *fact type*, *subspace*, *measure*, *breakdown*, and *focus*. In the definition, *fact type* indicates the category of information; *subspace* defines the data scope; *breakdown* divides the subspace into groups; *measure* calculates each group's numerical value; *focus* highlights the data items we need to pay attention to. For example, "California is the state that votes most for Biden" is a data fact whose *fact type* is *extreme*; it uses the data about Biden as *subspace*, the number of votes as *measure*, the states of the US as *breakdown*, and California as *focus*. As existing tools [SDES18, WSZ*19, SXS*20] already support the easy configuration of data facts from tabular data, this work assumes that our users already have a tabular dataset and a sequence of formulated data facts as input.

Mapping Facts to Clips. Given the data facts, AutoClips first maps each of the data facts to potential video clips (Fig. 2- step 1). Here, we define a clip as a visualization animated with a series of motions to convey a certain data fact. For example, one clip that can be used to present "the distribution of Biden's votes among states" is a bubble map where the bubbles representing the votes for Biden in different states gradually reveal. AutoClips makes such decisions by searching for the fact-clip mappings in the fact-driven clip library we manually built in Section 4. Note that in this step, a data fact can be mapped to multiple potential clips.

Composing the Clips. Next, AutoClips selects the optimal clip for each data fact and assembles the clips into a data video (Fig. 2- step 2). This step is achieved by an automatic algorithm (Section 5), which takes a set of design requirements into consideration. At last, AutoClips can output a complete data video given the input of a sequence of data facts.

3.3. Design Requirements

Below we describe a set of high-level design requirements to guide the two steps of AutoClips.

3.3.1. Clip Generation

Previous studies have suggested a set of design principles for creating animation for visualizations. For example, Tversky *et al.* [TMB02] proposed two high-level principles for animation design, including *congruence* and *apprehension*. Heer and Robertson [HR07] further broke down the two principles into specific design guidelines, such as avoiding ambiguity, using consistent semantic operators, and using staging for complex transitions. Based on these guidelines, we derived three design requirements for the clip generation in AutoClips.

R1 Ensure fact comprehension. A clip should communicate the corresponding data fact clearly and effectively. In other words, all the motions in the clip should be meaningful to support the presentation of the data fact.

R2 Improve fact recognizability. The clips that designed for a certain type of data fact should be distinguishable from the clips designed for another data fact. Viewers should be able to recognize what data fact a clip is presenting easily.

R3 Guide viewers' attention. A clip should use animation properly to guide viewers' attention and facilitate data perception. In accordance with [HR07], we use staged animation to present motions step by step.

3.3.2. Video Composition

We also surveyed literature that offered guidelines for how to create a narrative sequence. For example, Hullman *et al.* [HDR*13] proposed the idea of transition cost to measure the number of changes required to transform one visualization state to another; from the perspective of the audience, the transition cost should be minimized. When combining multiple visualizations, visual consistency, parallel narrative structure, and visualization diversity [HDR*13, QH17] are important factors that may influence the story perception. Moreover, for the production of videos, coherence [KY98] and duration [AHRL*15] also require careful manipulation. By referring to the above studies, we propose three design requirements for video composition.

R4 Optimize the visualization sequence. We should make sure that the visualizations chosen by AutoClips can be combined compatibly and facilitate storytelling. The resulting visualization sequence should be low in the transition cost, have good visual consistency and diversity, and can support the narratives.

R5 Enhance video coherence. The data videos generated by AutoClips should have smooth transitions between successive clips [Woo84] and support the effective presentation of narrative structures such as parallelism.

R6 Ensure proper video duration. The data videos generated by AutoClips should be compacted in duration as long as the animation is long enough present visual changes clearly [HR07].

4. A Fact-Driven Clip Library

This section introduces the fact-driven clip library we constructed. First, we describe the methodology used to build the library. Then, we present the fact-driven clips in the library in detail. Last, we describe how we implement the library with programming.

4.1. Methodology

According to our definition, a fact-driven clip in AutoClips is a visualization animated with a sequence of motions to convey a certain data fact. AutoClips only focuses on common statistical charts and maps. We did not include pictographs because pictograph is composed by icons whose shapes can be very diversified and are usually decided by the semantic meaning of data. Parsing the semantic meaning of data and generating corresponding icons are beyond the scope of this work.

To guide the design of fact-driven clips, we first have to know what visualizations and animations have been used to present data facts in the wild. Thus, we collected 230 high-quality data videos from online video platforms, including Youtube.com and Vimeo.com. We first searched for media channels that are famous for producing data videos, such as The Economist, TED-Ed, and Vox. We also searched keywords such as "data video", "animated infographic", and "motion infographic" to collect the data videos with most views. In line with [AHRL*15], we considered a data video qualified and included it into our corpus if it contains at least one data visualization.

Then, we watched the 230 data videos throughout and identified 1722 video clips that contain animated visualizations. Three of the authors labelled each of the clips from three aspects: (i) what data fact does the clip convey, (ii) what visualization has been used to present the data fact, and (iii) what animation has been used to present the visualization. For aspect (i), we considered ten most common data fact types mentioned by previous research [SXS*20], including *association*, *categorization*, *difference*, *distribution*, *extreme*, *outlier*, *proportion*, *rank*, *trend*, and *value*. For aspect (ii), we referred to existing visualization taxonomies [BVB*13, AHRL*15] to label the visualizations used in data videos. For aspect (iii), we labelled animation as a sequence of low-level motions. When naming the animations, we followed the concepts proposed by [ARL*16] (e.g., grow bars, highlight, display label). Newly observed animations were named by the authors through iterative discussions. Given that the amount of clips was large, the three authors first labelled 172 clips (10%) independently. Then, they met to compare their labels and discuss mismatches. After achieving consensus, they then labelled another 172 clips (10%) independently and met to discuss. After resolving disagreements and achieving final consensus, the three authors labelled the rest of the clips.

After the labeling, we used the frequency of the chart types and animations in each type of data fact as the criteria to design our clip library (see supplementary file for the analysis result). For instance, the most used visualization to present *extreme* is bar charts, and the commonly observed motions include "highlight", and "add annotation". Based on the analysis result, one author, who is a visualization designer, designed 38 prototype clips by referring to the top-ranked visualizations and animations used to present each data fact type in our corpus. During this process, the author also took our design requirements (**R1**-**R3**) into consideration. We also involved 12 professional users into the design process, including four animation designers, four data journalists, and four senior graduate students majored in data visualization. We presented them with the prototype clips we created, explained our design goals, and then asked for their suggestions. User feedback suggested that some data facts were similar with each other in nature (e.g., *extreme* and *outlier*), so the animation used to present such facts should be more distinguishable. We also removed two clips. For example, we dropped a clip which visualizes *trend* and *association* using an animated bubble chart because users mentioned that this clip conveys compound data facts and is significantly different from other clips. At last, we arrived at a final set of 36 clips. As shown in Fig. 3, our design finally covers 66.9% of the data visualizations used in the corpus.

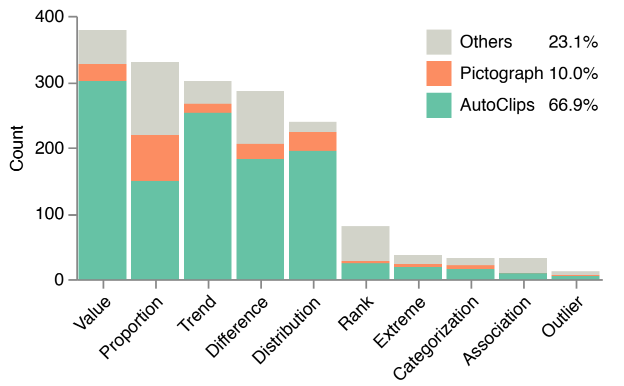

Figure 3: *AutoClips covers ten common data facts and 66.9% of the visualizations identified in our corpus.*

4.2. Library Design

Fig. 4 illustrates all clips in the fact-driven clip. The animated version of this library can be browsed at `https://autoclips.idvxlab.com`. The clips in the library are categorized according to their corresponding data fact types. Below we describe the fact-driven clip library in detail.

Association. This fact tells a relationship between two variables. In data videos, *association* is often visualized with scatter plots. To convey association, we first let the scatter points fade in, then draw a regression line from the left to right to show the relationship.

Categorization. This fact presents a set of nominal data. Categorization is often visualized with bubble charts, filled maps, and treemaps; different categories are often colored with different hues. To convey categorization, we use animation to reveal each colored category one by one.

Difference. This fact compares the difference between two data values. In data videos, *difference* is often visualized with maps, horizontal bars, texts, and vertical bars. The animation used to convey difference consists of two steps. First, we present the two objects that are going to be compared. Then, a number fades in to show the value of difference. For bar charts, we also draw reference lines pointing to their axes to facilitate value comparison.

Distribution. This fact assigns data values into a set of groups and presents the value of each group. *Distribution* is usually visualized with area charts, bubble charts, bubble maps, filled maps, horizontal bar charts, treemaps, and vertical bar charts. We use animation to grow all the groups all at once.

Extreme. This fact indicates the maximal or minimal value among various values. The display of *extreme* often occurs in area charts, bubble charts, line charts, texts, and vertical bar charts. To convey this fact, we use animation to first highlight the object (e.g., circle, bar, point) with the extreme value, then annotate the number beside the object.

Outlier. This fact points out the data points that significantly differ from others. In data videos, *outlier* often exists in line charts and bar charts. To show outlier data, We first draw a benchmark line to show the mean value of all the data values. Then, we highlight the outlier by drawing a red circle around the outlier or a red arrow pointing to the outlier. Also, to show the specialness of the outlier, all the other elements in the visualization fade to a lower opacity.

Proportion. This fact tells the part-to-whole relationship. In data videos, *proportion* is often visualized as donut charts, pie charts, progress bars, and treemaps. When animating *proportion*, we first present the visualizations, then highlight a certain part of the visualizations using a different color.

Rank. This fact presents a numerically ordered series. *Rank* is extensively visualized with horizontal bar charts. To convey *rank*, we first let all bars grow from the left to the right, then highlight the top-ranked items one by one.

Trend. This fact presents the tendency over a period of time. *Trend* is often visualized with area charts, bubble charts, line charts, and bar charts. We first use animation to gradually display temporal data from left to right according to the social convention that left denotes the past and right denotes the future. Then, we draw a line with an arrow to summarize the increase or decrease.

Value. This fact retrieves the exact value of a data attribute from the dataset. *Value* is often visualized with bubble maps, filled maps, horizontal bar charts, texts, and vertical bar charts. To display *value*, our animation first illustrates the corresponding objects and then show the number of the value.

4.3. Implementation

The fact-driven clip library was implemented in the JavaScript language using the D3.js framework [BOH11]. Each clip implements the visual encoding and a series of pre-defined motions independently. AutoClips uses the plug-in architecture, thus new clips can be easily extended to this library.

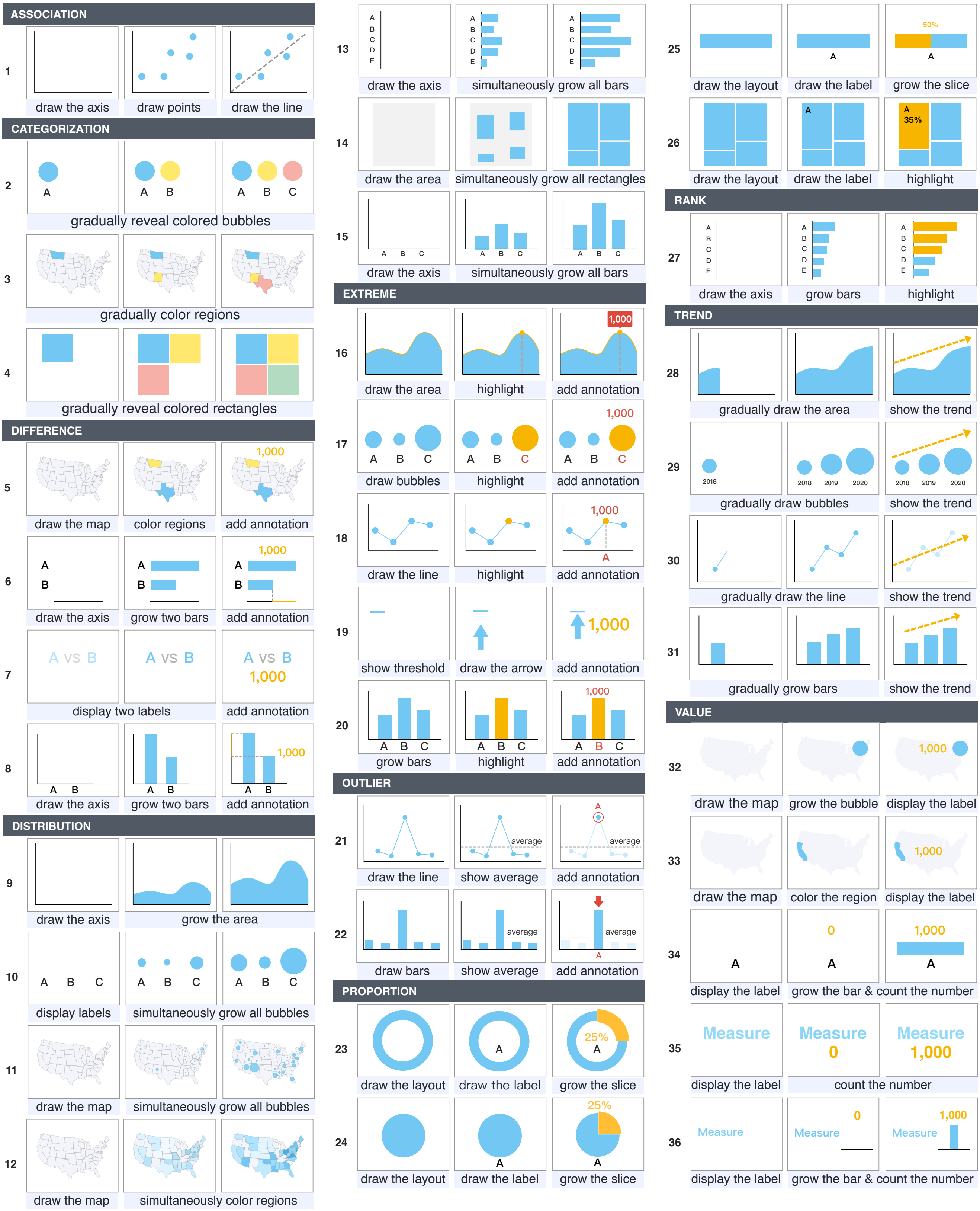

Figure 4: *The fact-driven clip library.*

5. An Algorithm for Data Video Generation

This section introduces our algorithm used to compose the clip candidates into a data video. To better explain the algorithm, we use a story about the 2020 US presidential election as our running example. The dataset presents the number of votes that each of the presidential candidates received in different US states. We invited a data journalist that we interviewed in Section 3 to create a data story based on this dataset. The resulting data story contains eight data facts, as described in Table 1.

Table 1: *The example story for algorithm demonstration*

ID	Fact Type	Fact Description
1	*value*	The sum of all votes in the US presidential election
2	*distribution*	The votes received by each presidential candidate
3	*extreme*	The maximum number of votes received by one candidate
4	*distribution*	The distribution of Biden's votes across all the states
5	*rank*	The votes for Biden by state sorted in descending order
6	*distribution*	The distribution of Trump's votes across all the states
7	*rank*	The votes for Trump by state sorted in descending order
8	*difference*	The difference in votes between Biden and Trump

Fig. 5 shows the workflow of our algorithm. In summary, the workflow consists of three steps: (i) clip selection, (ii) clip arrangement, and (iii) duration configuration.

5.1. Clip Selection

Since one data fact may be mapped to multiple potential video clips (in our library, there are on average 3.6 potential clips for one data fact, thus leading to 34,300 possible clip combinations for an eight-fact story), the algorithm first needs to select an optimal clip for each of the data facts. AutoClips uses an optimization-based method to automatically select a sequence of clips based on **R4**: $S = (c_1, c_2, \cdots, c_n), c_i \in \mathcal{L}$, where $\mathcal{L}$ is our fact-driven clip library. During the selection, we considered four computational features, including (i) the transition cost, (ii) the parallel narrative structure, (iii) the visual consistency, and (iv) the diversity of visualizations.

- ***Minimize the transition cost.*** Transition cost, according to [HDR*13], is the perception cost to track the visual changes between two consecutive visualizations. Since huge visual changes in the visualization sequence may burden the audience's cognition, researchers have suggested that the transition cost should be minimized. Accordingly, AutoClips minimizes the total transition cost for the clip sequence:
$$\tau = \sum_{i=1}^{n-1} cost(c_i, c_{i+1}) \tag{1}$$
where $cost(c_i, c_{i+1})$ is the sum of transition costs [KWHH17] of all single changes in visual representation between clip c_i and c_{i+1}, including modifying mark type and visual encodings. Therefore, AutoClips prefers similar visualizations for the consecutive facts (e.g., Fact 2 and 3 in Fig. 5-1).
- ***Preserve the parallel narrative structure.*** A parallel narrative structure adopts a repeated pattern to tell a story [HDR*13]. For example, in our example story, the storyteller first presents the *distribution* and *rank* of Biden's votes (Facts 4 and 5), then presents the *distribution* and *rank* of Trump's votes (Facts 6 and 7), thus constituting a parallel structure. AutoClips identifies the pattern P that repeated m times ($m > 1$) from the original data fact sequence by considering all types of changes, including the changes of fact type, measure, subspace, breakdown and focus. Then AutoClips gives priority to using uniform clips for the parallel sub-sequences $\{s_1, \ldots, s_m\}$.
$$\mathcal{P} = \frac{1}{|P|} \sum_{i=1}^{|P|} u(s_1(i), \cdots, s_m(i)) \tag{2}$$
where $|P|$ is the length of the sub-sequence; $s_j(i)$ indicates the ith clip in the sub-sequence s_j; $u()$ computes the uniformity of the function parameters. Thus, AutoClips tends to use the same clips to display each sub-sequence in a parallel structure (e.g., Facts 4-5 and Facts 6-7 in Fig. 5-1).
- ***Maintain visual consistency.*** Visual consistency requires using consistent visual representations across different views [BWK00]. It has been proved that maintaining visual consistency can make the visualizations easy-to-follow and reduce inaccurate inferences [QH17]. Formula 3 shows how AutoClips deals with visual consistency.
$$\mathcal{C} = \frac{1}{2n_t} \sum_{i=1}^{n_t} u(\mathcal{V}|t_i) + \frac{1}{2n_d} \sum_{i=1}^{n_d} u(\mathcal{E}|d_i) \tag{3}$$
where n_t and n_d indicate the number of fact types and data fields involved in the clip sequence; $u(x|y)$ computes whether x, a set of visualizations $\mathcal{V}$ or encodings $\mathcal{E}$, are totally uniform when given a specific y, which can be a fact type t_i or data field d_i. Therefore, AutoClips asks for using consistent chart type for the same fact type (first term) and consistent visual encoding for the same data field (second term). For example, when presenting the facts about *distribution*, AutoClips uses the bubble size to encode the number of votes (Facts 2, 4, 6 in Fig. 5-1).
- ***Improve the diversity of visualizations.*** In our library, some visualizations (e.g., bar charts) can be used to present multiple data facts. However, according to [WSZ*19], it is better to use different visualizations to present different data fact types so as to improve the discrimination among the data facts. AutoClips defines diversity $\mathcal{D}$ using the number of unique visualizations types divided by the total number of data fact types. Thus, AutoClips tries to use unique visualization types for different fact types. As depicted in the Fig. 5-1, the resulting story uses five different visualizations for the five input data fact types, thus making $\mathcal{D} = 1$.

Ultimately, the goal of our algorithm is to find a clip sequence that maximizes the satisfaction of all the above considerations. Therefore, we define the overall reward function $Reward(S)$ in Formula 4. We set default weights $\omega_1 = -1, \omega_2 = \omega_3 = \omega_4 = \frac{1}{3}$.

$$Reward(S) = \omega_1 \cdot \tau + \omega_2 \cdot \mathcal{P} + \omega_3 \cdot \mathcal{C} + \omega_4 \cdot \mathcal{D} \tag{4}$$

5.2. Clip Arrangement

Given the selected clips, AutoClips then arranges the clips into a coherent video sequence based on **R5**. To improve coherence, we considered two issues: how to smooth local transitions and how to better present parallel structures.

- ***Smooth Local Transitions.*** A local transition is the transition between two consecutive clips. For example, in our example, Fact 2 shows how many votes every candidate received, and Fact 3

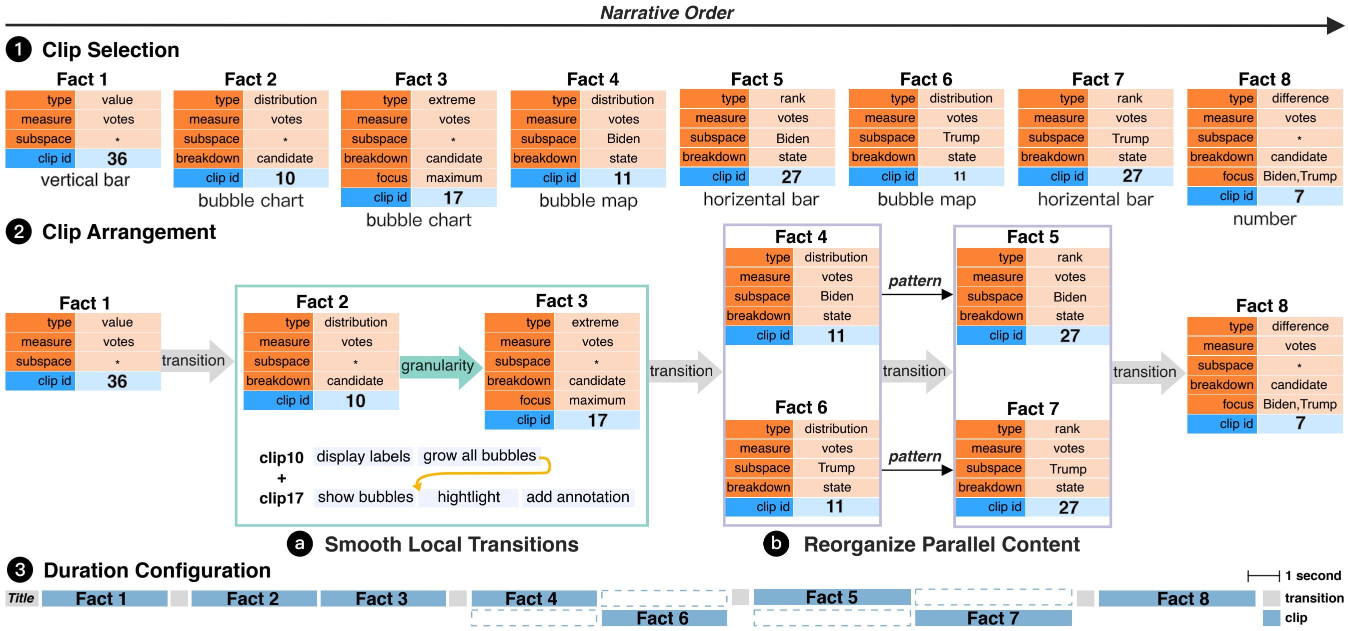

Figure 5: *The workflow of the algorithm, using the story of the 2020 US presidential election as an example. The algorithm takes a sequence of data facts as input and processes it in three steps: (1) clip selection, (2) clip arrangement, and (3) duration configuration.*

shows which candidate received the most votes. These two facts used a shared visualization and Fact 3 highlights a certain part of Fact 2. This transition, according to Hullman *et al.* [HDR*13], is called *granularity* transition. Apart from the *granularity* transition, AutoClips also detects the *temporal*, *comparison*, and *spatial* transitions [HDR*13] between clips. If the relationship between two clips falls into the four categories, AutoClips will merge the animation of the two clips and omit unnecessary motions. For example, before smoothing local transitions, the procedure of presenting Facts 2 and 3 should be: displaying the bubble chart (for Fact 2), removing the bubble chart, displaying the bubble chart (for Fact 3), highlighting and annotating the extreme, and removing the bubble chart. However, after smoothing local transitions, the animation becomes: displaying the bubble chart, highlighting and annotating the extreme, and removing the bubble chart (see Fig. 5-2a). If a transition does not fall into any of the four categories, AutoClips takes it as an *other* transition and simply use "dissolve" as the default transition effect.

- ***Reorganize Parallel Content.*** While parallelism is an extensively used strategy in storytelling [HDR*13, KWHH17], how to present parallel content with visualization remains a challenge. For example, in our example story, if we present the parallel content (Facts 4-7 in Fig. 5-2b) one by one, viewers need to watch the following four scenes: the bubble map of Biden, the bar chart of Biden, the bubble map of Trump, and the bar chart of Trump. Such arrangement has obvious drawbacks. First, presenting similar visualizations separately is not efficient and will increase viewers' cognitive load to read the visualizations. Second, presenting parallel content one by one also hinders effective data comparison since viewers tend to forget the content of former clips very fast [IXTO11]. Thus, to facilitate the presentation of parallel content, AutoClips reorganizes the sub-sequences within the parallel structure and juxtaposes two visualizations supported by the same type of data fact in one scene. Thus, as depicted in Fig. 5-2b and Fig. 1-5, AutoClips arranges Facts 4 and 6 into one scene so that viewers can watch Biden's and Trump's vote map simultaneously; then, AutoClips arranges Fact 5 and 7 into another scene so that viewers can compare the top states that voted to the two candidates most. For now, AutoClips detects all possible parallel sequences but only selects two significant ones to display. They are placed side by side and played in sequence.

5.3. Duration Configuration

Last, AutoClips configures the duration of the clips to ensure an optimal video length according to **R6**. Three issues have been considered: First, we want the data video to be condensed as a whole since a too-long presentation time will disengage the viewers [AHRL*15]. Second, the animation should not be too fast; Robertson *et al.* [RCCR02] have recommend that each animated transition should last around one second to enable people to perceive the visual changes. Third, people usually want to pay more attention to important facts when watching videos. Thus, important facts should have a longer duration. Based on these considerations, AutoClips configures the duration using linear programming:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} T(c_i) \\ \text{subject to} \quad & \forall T(c_i) > T(c_j) \qquad \text{if } I(f_i) > I(f_j) \\ & \forall T(c_i) \geq \min T(c_i) \end{aligned}$$

where $T(c_i)$ is the duration of the ith clip, and the $I(f_i)$ is the importance score of the ith fact. The $I(f_i)$ can be set manually or be automatically calculated using existing statistical techniques [THY*17]. As shown by Fig. 5-3, the final duration of our example story is 38 seconds, and some facts have been allotted more time than others. The final generated video is shown in Fig. 1.

6. Evaluation

In this section, we first present two user studies which evaluate (i) the clips and (ii) the data videos generated by AutoClips respectively and then present an example gallery.

6.1. Study I: Evaluating the Clips

To assess whether the clips we proposed in the fact-driven clip library can convey data facts correctly, we conducted a study with 30 participants. The participants were presented with all the clips in the library and were asked to rate the clips from three aspects, including recognizability, comprehensibility, and engagement level.

Procedure. We recruited 30 participants (18 female) aged between 22 and 28 years old ($M = 23.6, SD = 1.63$) from a visualization course in a college. All participants declared that they have knowledge in visualization. At the beginning of the study, we introduced the definition of data facts to the participants and explained the ten data fact types with examples. Then, we presented the 36 clips in our library to the participants one by one. We used a Latin square to display the clips randomly. When viewing each of the clips, the participants were asked to recognize what data fact did the clip present and to report how confident they felt about their answers using a 5-point Likert scale (1 denotes totally not confident, 5 denotes totally confident). Besides, they were asked to rate to what degree did they agree that the clip is comprehensible and engaging, respectively, using a 5-point Likert scale (1 denotes strongly disagree, 5 denotes strongly agree). During the study, the participants were allowed to replay the clips and leave comments. The study typically lasted 25.5 minutes for each participant.

Result. We first analyzed the accuracy of the recognition task. Overall, 79.5% of the answers were correct and the average confidence level was 4.20 ($SD = 0.96$). As shown in Fig. 6, among the ten data facts, *value*, *difference*, and *proportion* earned the highest accuracy; almost all participants have recognized these facts successfully. On the contrary, the clips of *distribution* and *extreme* were thought less recognizable. User feedback showed that confusion mainly occurred between *distribution* and *categorization* (e.g., "*I thought bar charts also present categories*"), *distribution* and *difference* (e.g., "*these two data facts are similar at a glance*"), as well as between *extreme* and *value* (e.g., "*I guess more background information may help me distinguish extreme and value better.*"). We found that there were three major reasons that led to such confusion. First, some participants were new to the concept of data facts. Although we had presented them with the definitions of data facts at the beginning of the study, the participants tended to made mistakes on data facts that have similar appearances. Second, the participants' prior knowledge about visualization may also mislead their choices. Third, this study presented the clips individually to the participants rather than putting them into story contexts, thus inevitably increasing the difficulty of fact recognition. In general, given that most clips were correctly recognized and that confusion was mostly caused by external factors, we thought that our clip library was recognizable. Also, based on user feedback, we slightly adjusted the design of two clips to refine our library further (i.e., adding an annotation beside the reference lines in clips 21 and 22 to show that the lines represent the averages).

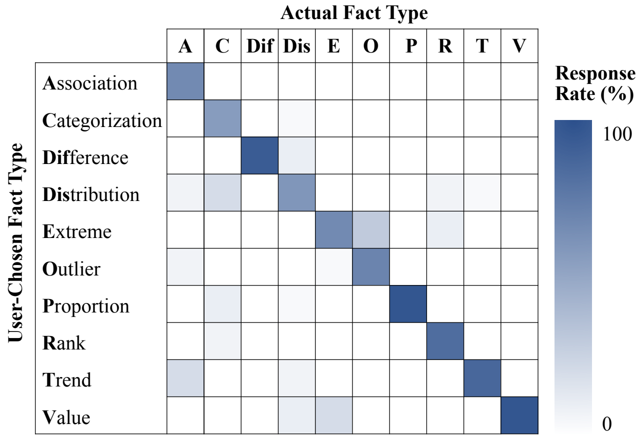

Figure 6: *The recognizability response in Study I. Columns list the actual fact type of all clips in the library while rows show how the participants interpreted the fact types of the clips. Correct responses are located along the diagonal of the table.*

We then analyzed the ratings for comprehensibility and engagement level. Overall, the participants thought that the clips in our library were comprehensible ($M = 4.17, SD = 1.05$) and engaging ($M = 3.47, SD = 0.74$). For example, one participant commented that *"The animation helps me understand the data fact better; I like the trendline you draw to summarize the increasing trend."* The staged animation was favored by many participants (e.g., *"I love how you break down the animation into multiple steps so that I can follow it easily"*). Some participants also expressed their overall appreciation for our library (e.g., *"It is interesting to know that even for the same chart type, different animation design can deliver different data insights"*).

6.2. Study II: Evaluating the Data Videos

In this section, we involved real world stories to evaluate the data videos generated by AutoClips. We prepared six different data stories with different topics (US presidential election (4×19843), Marvel Universe (8×12443), Amazon bestsellers (7×549), business investment (6×1234), COVID-19 (5×1494), and brand marketing (4×275)) and created four versions of data videos for each of the story, including one data video generated by AutoClips, one data video manually assembled by human designers based on our clip library, and two data videos whose visualizations and animations were randomly picked from the library. It should be noted that the designer's manual process is based on our clip library. Thus, we can control all the factors like design styles to make a fair comparison. The four versions of data videos all used the same storyline (i.e., an identical sequence of data facts). In total, we prepared 24 data videos for the six data stories. We then recruited 24 participants to watch all the data videos and rate the videos in terms of comprehensibility and engagement level.

Procedure. Twenty four participants (15 females) aged from 21 to 33 years old ($M = 23.96, SD = 2.49$) participated in this study. 20 participants had experience in visualization, 16 participants had experience in data analysis, and 14 participants had experience in an-

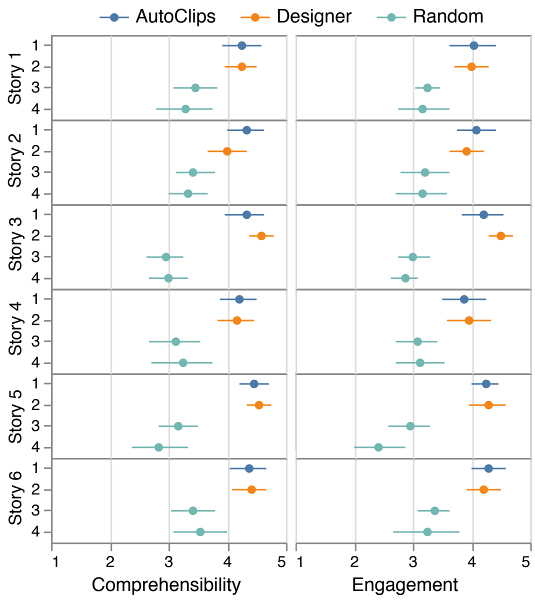

Figure 7: *Mean ratings and 95% confidence intervals per story.*

imation or video production. The study employed a within-subject design. First, we introduced the concept of data video to the participants and explained our research intent. Then, the participants entered into the story page where we first presented a synopsis of the story and then presented the four versions of data videos corresponding to the story to the participants in random order. After watching the four videos, the participants were asked to rate whether each of the videos was comprehensible and engaging using a 5-point Likert scale. Also, the participants were asked to write down reasons to explain their ratings. We iterated this process until all the six stories and 24 data videos have been rated by the participants. During the study, the participants were allowed to replay the videos. The study lasted 50 minutes on average for each participant.

Result. Fig. 7 shows the participants' ratings for the 24 data videos. Obviously, randomly-generated data videos earned low scores in all the six stories, and the data videos generated by AutoClips had comparable scores with the videos manually created by human designers. Among the six stories, AutoClips outperformed human designers in stories 2 and almost equalled human designers in stories 1, 4, 5 and 6; it performed a bit worse than human designers in story 3. Overall, AutoClips was rated high in all stories. Statistics also yielded similar results. Since the ratings were not normally distributed, we ran the non-parametric Friedman tests within each story to compare the four versions of videos. Results showed that for all the stories, there was a significant difference among the four videos ($p < 0.05$). The post-hoc pairwise comparisons (a Holm–Bonferroni correction was applied) further indicated that the videos generated by AutoClips had significantly higher ratings in comprehensibility and engagement level than randomly-generated videos ($p < 0.05$). Meanwhile, there was no significant difference between the videos generated by AutoClips and by human designers, suggesting that AutoClips can generate data videos comparable to human work.

Qualitative feedback showed that the participants tended to set higher scores to data videos with higher visual consistency (e.g., "*I like this video best (produced by AutoClips) because it uses similar charts to present similar things, so I can grasp the visuals very fast*"). Some participants also appreciated the juxtaposition of parallel content (e.g., "*I gave this video (produced by AutoClips) the highest score because it puts two similar charts into one scene, and this is an efficient way to compare data*"). On the contrary, the participants set lower scores to data videos with inconsistent visualizations or unsmooth transitions (e.g., "*This video (produced randomly) is poorly designed. I think the first and the second charts should be merged together?*"). These comments showed that our algorithm design is effective. We also collected some constructive user feedback which will guide our future work. For example, some participants said that the videos presented were relatively simple; more visualization types, embellishments, and textual descriptions could be added in the future.

6.3. Sample Videos

To demonstrate the expressiveness of AutoClips, we generate a variety of example videos based on a set of public datasets. We put these examples on an online video gallery at https://autoclips.idvxlab.com. We can see that AutoClips can automatically produce data videos with a reasonable level of diversity.

7. Conclusion and Future Work

This work presents AutoClips, an automatic approach to generating data videos from a sequence of data facts. We first built a fact-driven clip library by analyzing 230 data videos online and conducting interviews. Next, we constructed an algorithm which generates data videos from data facts through three steps: clip selection, clip arrangement, and duration configuration. Last, we evaluated the effectiveness of AutoClips via two controlled user studies. The results showed that AutoClips can generate comprehensible and engaging data videos which have comparable quality with human-made videos, indicating that AutoClips can significantly lower the barrier of the creation of data videos.

There still remains a great deal of future research. First, we simplified and reduced the visual design space of a clip in this work so that the paper can focus on the generation algorithm. Especially, we treated different visualizations equally during clip selection. However, the perceptual effectiveness of each chart type could be another important factor to present a data fact, which needs more future research. Besides, further exploring the design space and taking other elements such as embellishments, chart transitions into consideration may also greatly improve the video representation. Second, AutoClips currently only supports tabular data and favors datasets with diverse column types. This data scope also limits the potential visual representation of the data video. Continued research is needed to include diverse data types, such as graphs, spatio-temporal data, or textual data.

Acknowledgement Nan Cao is the corresponding author. This work was supported in part by the National Natural Science Foundation of China 62072338, 6200070909, and NSF Shanghai 20ZR1461500.

References

[Ado] ADOBEAFTEREFFECTS CC: Adobe after effects. https://www.adobe.com/products/aftereffects.html. [Online; accessed 4-Nov-2020]. 2

[AHRL*15] AMINI F., HENRY RICHE N., LEE B., HURTER C., IRANI P.: Understanding data videos: Looking at narrative visualization through the cinematography lens. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015), pp. 1459–1468. 1, 2, 3, 4, 8

[APP10] ARCHAMBAULT D., PURCHASE H., PINAUD B.: Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE Transactions on Visualization and Computer Graphics 17*, 4 (2010), 539–552. 3

[ARL*16] AMINI F., RICHE N. H., LEE B., MONROY-HERNANDEZ A., IRANI P.: Authoring data-driven videos with dataclips. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2016), 501–510. 1, 2, 4

[ARL*18] AMINI F., RICHE N. H., LEE B., LEBOE-MCGOWAN J., IRANI P.: Hooked on data videos: assessing the effect of animation and pictographs on viewer engagement. In *Proceedings of the International Conference on Advanced Visual Interfaces* (2018), pp. 1–9. 1, 2, 3

[BLIC19] BREHMER M., LEE B., ISENBERG P., CHOE E. K.: A comparative evaluation of animation and small multiples for trend visualization on mobile phones. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2019), 364–374. 3

[BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2301–2309. 2, 3, 5

[BVB*13] BORKIN M. A., VO A. A., BYLINSKII Z., ISOLA P., SUNKAVALLI S., OLIVA A., PFISTER H.: What makes a visualization memorable? *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2306–2315. 4

[BWK00] BALDONADO M., WOODRUFF A., KUCHINSKY A.: Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (2000), pp. 110–119. 7

[CRP*16] CHEVALIER F., RICHE N. H., PLAISANT C., CHALBI A., HURTER C.: Animations 25 years later: New roles and opportunities. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (2016), pp. 280–287. 3

[EDF08] ELMQVIST N., DRAGICEVIC P., FEKETE J.-D.: Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1539–1148. 3

[Fis10] FISHER D.: *Animation for Visualization: Opportunities and Drawbacks*, beautiful visualization ed. O'Reilly Media, 2010. 2

[GZL*20] GE T., ZHAO Y., LEE B., REN D., CHEN B., WANG Y.: Canis: A high-level language for data-driven chart animations. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 607–617. 2, 3

[HDR*13] HULLMAN J., DRUCKER S., RICHE N. H., LEE B., FISHER D., ADAR E.: A deeper understanding of sequence in narrative visualization. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2406–2415. 4, 7, 8

[HR07] HEER J., ROBERTSON G.: Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1240–1247. 2, 3, 4

[IXTO11] ISOLA P., XIAO J., TORRALBA A., OLIVA A.: What makes an image memorable? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2011), pp. 145–152. 8

[Jef13] JEFFS P.: This viral video is right: We need to worry about wealth inequality. https://www.washingtonpost.com/, 2013. 1

[KH20] KIM Y., HEER J.: Gemini: A grammar and recommender system for animated transitions in statistical graphics. *IEEE Transactions on Visualization and Computer Graphics* (2020). 2, 3

[KWHH17] KIM Y., WONGSUPHASAWAT K., HULLMAN J., HEER J.: Graphscape: A model for automated reasoning about visualization similarity and sequencing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (2017), pp. 2628–2638. 7, 8

[KY98] KENDER J. R., YEO B.-L.: Video scene segmentation via continuous video coherence. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)* (1998), IEEE, pp. 367–373. 4

[Ltd] LTD K. E.: Flourish. https://flourish.studio/. [Online; accessed 4-Nov-2020]. 2

[LWY*20] LU J., WANG J., YE H., GU Y., DING Z., XU M., CHEN W.: Illustrating changes in time-series data with data video. *IEEE Computer Graphics and Applications 40*, 2 (2020), 18–31. 2

[Pol12] POLITIZANE: Wealth inequality in america. https://www.youtube.com/watch?v=QPKKQnijnsM, 2012. [Online; accessed 4-Nov-2020]. 1

[PR] PEDERSEN T. L., ROBINSON D.: A grammar of animated graphics | gganimate. https://gganimate.com/. [Online; accessed 4-Nov-2020]. 2

[QH17] QU Z., HULLMAN J.: Keeping multiple views consistent: Constraints, validations, and exceptions in visualization authoring. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2017), 468–477. 4, 7

[RCCR02] ROBERTSON G., CAMERON K., CZERWINSKI M., ROBBINS D.: Animated visualization of multiple intersecting hierarchies. *Information Visualization 1*, 1 (2002), 50–65. 8

[RFF*08] ROBERTSON G., FERNANDEZ R., FISHER D., LEE B., STASKO J.: Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1325–1332. 2

[SDES18] SRINIVASAN A., DRUCKER S. M., ENDERT A., STASKO J.: Augmenting visualizations with interactive data facts to facilitate interpretation and communication. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2018), 672–681. 3

[SH10] SEGEL E., HEER J.: Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1139–1148. 1, 2, 3

[SXS*20] SHI D., XU X., SUN F., SHI Y., CAO N.: Calliope: Automatic visual data story generation from a spreadsheet. *IEEE Transactions on Visualization and Computer Graphics* (2020). 3, 4

[THY*17] TANG B., HAN S., YIU M. L., DING R., ZHANG D.: Extracting top-k insights from multi-dimensional data. In *Proceedings of the 2017 ACM International Conference on Management of Data* (2017), pp. 1509–1524. 8

[TLLS20] THOMPSON J., LIU Z., LI W., STASKO J.: Understanding the design space and authoring paradigms for animated data graphics. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 207–218. 2

[TMB02] TVERSKY B., MORRISON J. B., BETRANCOURT M.: Animation: can it facilitate? *International journal of human-computer studies 57*, 4 (2002), 247–262. 4

[TYT*20] TANG J., YU L., TANG T., SHU X., YING L., ZHOU Y., REN P., WU Y.: Narrative transitions in data videos. *2020 IEEE Visualization Conference (VIS)* (2020), 151–155. 2

[Woo84] WOODS D. D.: Visual momentum: a concept to improve the cognitive coupling of person and computer. *International Journal of Man-Machine Studies 21*, 3 (1984), 229–244. 4

[WSZ*19] WANG Y., SUN Z., ZHANG H., CUI W., XU K., MA X., ZHANG D.: Datashot: Automatic generation of fact sheets from tabular data. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2019), 895–905. 2, 3, 7