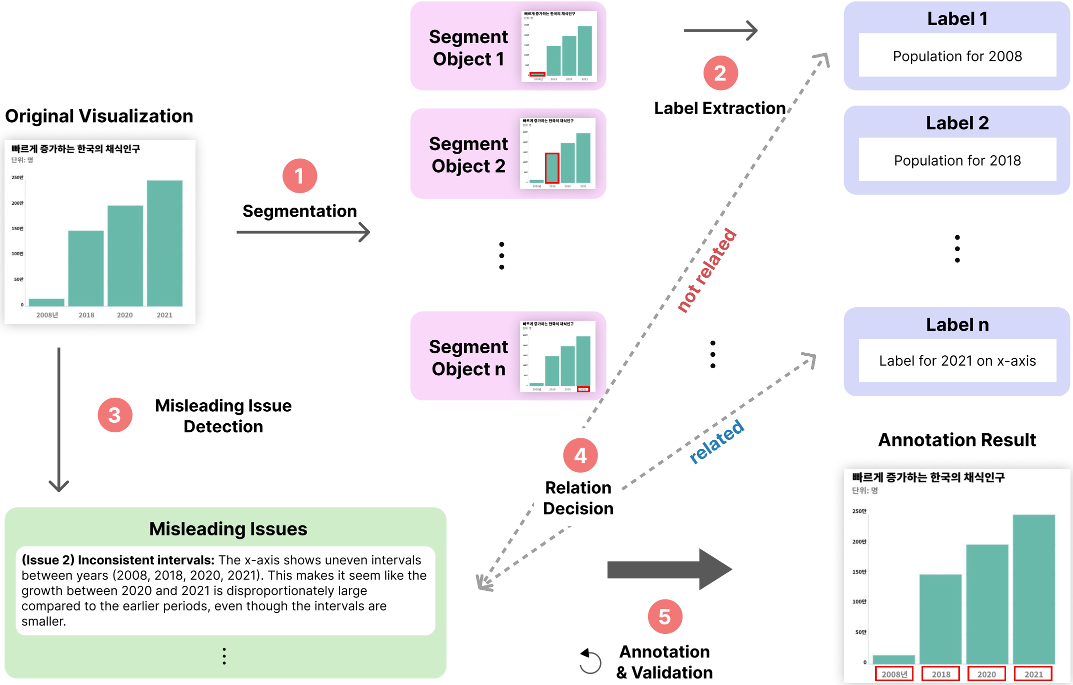
Automated Pipeline for Detecting and Analyzing Misleading Visual Elements

Min Hyeong Kim* Yumin Song* Yungun Kim* Aeri Cho* Soohyun Lee* Hyeon Jeon* Jinwook Seo†

Seoul National University



(Issue 2) Inconsistent intervals: The x-axis shows uneven intervals between years (2008, 2018, 2020, 2021). This makes it seem like the growth between 2020 and 2021 is disproportionately large compared to the earlier periods, even though the intervals are smaller.

Figure 1: The overview of our pipeline for visual annotation creation. (1) The pipeline begins by segmenting the input visualization into its constituent components. (2) Next, it generates labels in natural language for each segmented object. (3) The pipeline then detects and identifies several misleading issues in the input visualization. (4) To align the segmented objects with the identified issues, the pipeline assesses the relevance between each segment and its corresponding issues. (5) Finally, the pipeline validates the detected issues and their relevance to the objects.

Abstract

Data visualizations can sometimes misrepresent the underlying data, leading to *misleading* interpretations. However, existing systems fail to precisely identify which parts of a visualization contribute to misleading interpretations, leaving users uncertain about the misalignments. To address this issue, we develop a pipeline that automatically identifies the misleading parts within a visualization. Given an image file, our pipeline first detects graphical components of the visualization, converting them into structured objects. We then apply an algorithm to pinpoint misleading objects and explain how they contribute to distortions in interpretation. Our user study confirms that our pipeline accurately identifies misleading visualization designs, outperforming previous baselines. We also find that our pipeline supports participants in developing revision strategies to improve misleading visualizations.

Index Terms: Misleading Visualizations, Visual Annotation

1 Introduction

Visualizations convey information, intention, and messages. However, they can often be *misleading*. It is thus crucial for users to recognize whether a given visualization is misleading or not to correctly interpret the visualization. Various guidelines have been proposed to determine whether a visualization is misleading [28, 29]. Based on these guidelines, previous studies have attempted to detect misleading visualization designs [4, 5, 7, 15].

However, existing methodologies to detect and critique misleading visualizations cannot reveal which parts of visualizations are misleading. For example, checking whether given visualizations align with existing guidelines often returns exhaustive lists that are difficult to consume [4], leaves interpretation to the reader [15], or is applicable to only a limited set of issues and visualizations [5, 7]. This presents challenges for users trying to understand why visualizations are misleading and how to resolve them.

To address this problem, we propose a pipeline that automatically identifies which parts of the visualization design are misleading based on insights from a preliminary study. Our pipeline first objectifies visualization design components using image segmentation models, then generates descriptions on misleading visualization designs using multimodal large language models. These descriptions are then connected to the corresponding design components and presented to users through visual annotations. Users

*e-mail: {mhkim, ymsong, ygkim, archo, shlee, hj}@hcil.snu.ac.kr
†e-mail: jseo@snu.ac.kr, corresponding author

can then investigate design problems by examining the annotations overlayed on the original visualization. This pipeline fosters enhanced awareness of misleading visualization designs, enabling users to critically evaluate the presented information.

We evaluate our pipeline through a user study in which users were required to detect misleading design issues in a given visualization and proposed appropriate fixes. The results of our study verify that the visual annotation generated by our pipeline helps users identify misleading design issues more effectively and develop more robust and effective solutions. Compared to text-only systems, users identified issues more accurately and proposed more suitable fixes when aided by visual annotations. Users also reported that visual annotations helped them distinguish insignificant or false issues, improving both their task efficiency and overall experience.

In summary, our main contributions are as follows:

- We propose a pipeline that combines image segmentation and multimodal language models to identify misleading design elements in visualizations.
- We empirically demonstrate the effectiveness and accuracy of our pipeline in identifying misleading designs.
- We show that our pipeline helps users better understand and address design issues.

2 Related Work

Our work is relevant to two research fields: misleading visualizations and automated visualization evaluation.

2.1 Misleading Visualizations

Visualizations are an effective medium for delivering messages, but they can also easily mislead users. Several studies have explored how visualizations can mislead audiences, identifying factors such as deceptive design components [18, 19, 27], human preexisting beliefs [30, 31], and errors propagating throughout the visual analytics process [16]. Building on these previous avenues, Lo et al. [12] constructed a taxonomy of misleading visualizations and revealed that most errors occurred in the visualization design and plotting stages. Similarly, Lan et al. [11] categorize design flaws that can undermine a visualization's purpose; focusing on these issues from a public's perspective easily meets misleading visualizations.

Although these studies guide visualization designers to avoid misleading visualizations, such visualizations persist due to blunders or mischief [23]. To address this, our research proposes an automated pipeline that detects misleading components and enhances understanding through visual annotations and reasoning, which offer a more accessible way to follow existing guidelines.

2.2 Automated Visualization Evaluation

Automated evaluations for visualizations has been widely studied in literature [6]. These automated approaches provide benefits to visualization designers in creating more effective and reliable visualizations [9, 14, 17].

One common way to automatically evaluate visualizations is to integrate guidelines into the visualization design process. Visualization grammar such as *Vega-Lite* [24], which enables direct access to marks and channels, allows the integration of an additional layer checking if the specification fits all criteria in a predefined guidelines [3, 7]. Machine learning methods have also been adopted for visualization evaluation. Jung et al. [8] proposes workflows for data extraction from visualization by automatic chart type detection with user interaction. Subsequently, models such as CNN [20] and Mask R-CNN [10] made data extraction possible without user input.

Recently, Large Language Models (LLMs) have been proven to be effective in analyzing and evaluating visualizations. Shin et al. [25] developed a system to support the iterative design process of visualizations by delivering visualization feedback using LLMs. Moreover, Lo and Qu [13] confirmed that LLMs can effectively identify misleading components in visualizations correctly by testing various prompting methods and LLM models. Similarly, Alexander et al. [1] showed that GPT-4 models can well detect misleading visualizations.

These studies focus on evaluating visualization designs to help designers build better visualizations. However, despite such efforts, numerous poorly designed visualizations exist in the wild. We thus focus on helping *users* identify and understand such visualizations effectively. Our approach reduces the need for users to manually compare guidelines with visualizations by providing direct annotations that allow users to easily identify issues and access corresponding explanations across various chart types. Our approach thus contributes to informing users to become more active and critical in interpreting visualizations.

3 Preliminary Study for Prompt Design

We utilized the GPT-4o model to detect misleading visualizations, noting that its detection performance varies significantly depending on the prompt [13]. Therefore, we conducted a preliminary study to determine the optimal prompt configuration for our final pipeline. Referring to prior studies, we evaluated their performance based on how accurately each prompt identified misleading factors. A combination of Lo and Qu's dataset and manually collected visualizations from online news articles was used for the preliminary study.

Preliminary Study. Drawing inspiration from the works of Lo and Qu [13] and Alexander [1], we tested three prompts:

- Prompt #1: a guideline-based prompt (split chain of thoughts approach selected among prompts from Lo and Qu due to fine performance)
- Prompt #2: "Does this visualization from a news article depict data truthfully or misleadingly? Answer 'truthful' or 'misleading'. Name the issues and explain your answer." (a naive zero-shot prompt without additional examples or guidelines from Alexander [1])
- Prompt #3: Prompt #1 + "Capture any other issues outside the guideline if exist." (a combination of a guideline-based prompt and an open-ended question)

In our study, we followed a three-step process to evaluate the prompts. First, each prompt was applied to the same 50 images to identify visualization issues. Next, three authors independently assessed each issue, classifying them into one of three categories: (1) **True**: issues corresponded to genuinely misleading elements. (2) **False**: irrelevant or incorrect issues. (3) **Insignificant**: issues that were technically correct but had minimal or negligible impact. This includes items such as duplicated issues, issues regarding data source, context, or credibility, and color schemes in monochrome environments. Finally, three authors reached a consensus to finalize the classifications and ensure consistency in the results.

Table 1 presents the results of this preliminary study. According to the results, Prompt #2 outperformed the other prompts by identifying genuinely misleading elements over 40% more effectively and achieving the highest true rate. While Prompt #3 had the lowest number of false cases, its performance was not significantly different from Prompt #2 and had fewer true cases. Prompt #1 exhibited an overwhelmingly high proportion of insignificant results, which could increase cognitive load, and its performance in other metrics was also subpar. Therefore, as Prompt #2 demonstrated the best performance under our evaluation criteria, we incorporated this prompting technique into our pipeline.

4 Pipeline

We demonstrate our pipeline to identify which parts of visualizations are misleading. We first explain the input and output, and describe the pipeline architecture.

Table 1: Results of the preliminary study: how many true/false issues each prompt found. Prompt #2 (a naive zero-shot prompt) achieved the highest performance with the largest proportion of true issues identified.

Prompt	True	False	Insignificant	Total	True(%)
Prompt #1	76	28	146	**250**	30.4
Prompt #2	**123**	19	39	181	**68.0**
Prompt #3	86	**16**	37	139	61.9

Table 2: Order of sessions for user groups A, B, and C. The systems and image sets were ordered according to a 3×3 Latin square. *Lo*, *ours(text)* and *ours(ann)* indicate text results from Prompt #1, our system without visual annotations, and our full system with annotations, respectively.

Group	Session 1		Session 2		Session 3	
	System	Set	System	Set	System	Set
A	***ours(ann)***	S1	*ours(text)*	S2	*Lo*	S3
B	*Lo*	S2	***ours(ann)***	S3	*ours(text)*	S1
C	*ours(text)*	S3	*Lo*	S1	***ours(ann)***	S2

4.1 Input and Output

Our annotating pipeline for visualization analysis takes a visualization as an input in the format of bitmap images. The pipeline then outputs a list of descriptions on misleading visualization designs and visual annotations that denotes which parts of visualizations have the design problem. The output is provided in JSON format and includes descriptive labels and bounding boxes. These bounding boxes are later overlaid on the images for component highlighting.

Note that we decided to deliver misleading designs with visual annotations because previous studies have acknowledged the importance of providing feedback directly on errors instead of a separate natural language description [2, 7]. However, annotating text directly to visualizations can introduce additional bias depending on how it is presented [26] and may confuse users by blending with existing chart information. To avoid these issues, we only annotated the boundaries of the misleading parts of visualizations and provided descriptions separately, especially because boundaries are effective in delivering areas of interest [21].

4.2 Construction

The proposed system is structured as a modular pipeline to detect, annotate, and critique misleading visualizations while assisting users in identifying potential biases and alternative interpretations. The pipeline consists of five main stages: (1) segmentation, (2) label extraction, (3) misleading issue detection, (4) relation decision, and (5) annotation and validation. Each stage contributes to the systematic breakdown, analysis, and evaluation of visual components. The overall structure of this pipeline is shown in Figure 1.

Step 1: Segmentation of Original Visualization. The pipeline begins with the segmentation of an input visualization into components such as axis labels, data bars, legends, and gridlines. Segmentation is achieved through the pre-trained SAM2 [22] model. The system extracts bounding boxes to isolate specific regions and convert each segment into a base64-encoded image for further analysis. This step enables a fine-grained decomposition of the visualization, ensuring that each visual element is ready for independent evaluation.

Step 2: Label Extraction in Natural Language. After segmentation, the system generates natural language descriptions for each segment using GPT-4o, chosen for its performance in detecting misleading visualizations [1, 13]. These descriptions explain the function of each visual component, such as interpreting an axis tick label as "Label for 2021 on x-axis." This automated process eliminates manual effort while clarifying the relationships between data elements.

Step 3: Misleading Issue Detection. At this stage, the system identifies misleading issues within the visualization. GPT-4o is iteratively queried to critique the visualization and explain how individual components contribute to these issues, using Prompt #1 from the preliminary study detailed in section 3 .

Step 4: Relation Decision. The system determines the relevance of each segment to potential misleading issues. Using natural language reasoning, the language model classifies segments as either related or irrelevant. This filtering step ensures the pipeline focuses on the most relevant components and ensures that irrelevant visual elements do not obscure the analysis.

Step 5: Annotation and Validation. In the final step, the system validates the detected issues and their corresponding segments to ensure accuracy. False positives are filtered out through additional reasoning prompts, leaving only the critical misleading components. The outputs include an annotated visualization with visually highlighted misleading components and textual explanations describing the identified issues. This combined output enhances transparency and provides actionable insights into the visualization's biases.

Specific prompts and implementations are available in OSF[1].

4.3 Core Features

The two main features of our pipeline design are to give misleading issues and to ensure text-match-annotation control.

F1. Give Misleading Issues. The system provides misleading issues to clarify misleading or ambiguous elements of the visualizations. These issues act as corrective explanations, offering users a clearer and more accurate interpretation of the data. By providing these critiques, the system not only identifies problematic areas but also educates users on how to interpret the visual elements more critically. This approach encourages users to move beyond passive consumption of visualizations and engage more thoughtfully with the data.

F2. Text-match-Annotation Control. To guide users in understanding the misleading issues through direct annotation on visualizations, the system must ensure precision and clarity in aligning annotation elements with specific issues. To achieve this, the system incorporates a text-match annotation control, enabling users to overcome challenges in connecting textual critiques with corresponding visual elements. For example, if the system critiques a narrow vertical range that exaggerates trends, the corresponding annotation will precisely highlight the affected part of the chart. This interaction, such as highlighting the specific issue when hovering, helps users easily identify the visual element associated with the misleading issue, enhancing their comprehension and exploration.

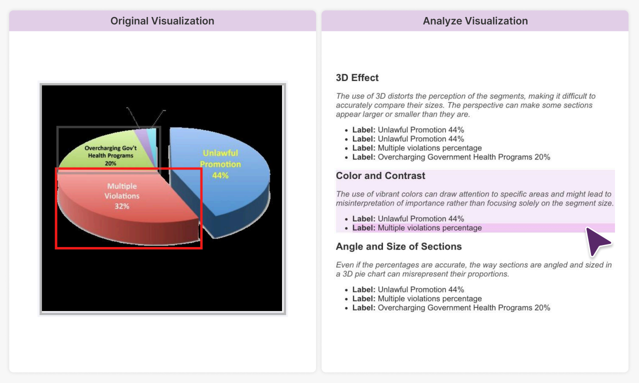
These core features are implemented through the modular design of our pipeline. The segmentation process enables a granular analysis of individual components, ensuring that subtle yet significant biases are detected. Also, by leveraging language models for description and reasoning, the system automates both semantic understanding and issue detection, minimizing manual intervention. The annotated outputs provide an interpretable critique of misleading visualizations, empowering users to critically evaluate the presented data.

5 User Study

5.1 Objectives and Design

We aim to verify the effectiveness of our system by comparing how users (1) identify the factors that make visualizations misleading and (2) attempt to fix those issues with three different interface for reporting visualization issues.

[1] https://osf.io/dn6gs/



3D Effect

The use of 3D distorts the perception of the segments, making it difficult to accurately compare their sizes. The perspective can make some sections appear larger or smaller than they are.

- **Label:** Unlawful Promotion 44%
- **Label:** Unlawful Promotion 44%
- **Label:** Multiple violations percentage
- **Label:** Overcharging Government Health Programs 20%

Color and Contrast

The use of vibrant colors can draw attention to specific areas and might lead to misinterpretation of importance rather than focusing solely on the segment size.

- **Label:** Unlawful Promotion 44%
- **Label:** Multiple violations percentage

Angle and Size of Sections

Even if the percentages are accurate, the way sections are angled and sized in a 3D pie chart can misrepresent their proportions.

- **Label:** Unlawful Promotion 44%
- **Label:** Multiple violations percentage
- **Label:** Overcharging Government Health Programs 20%

Figure 2: The experiment interface for our full system. The left panel displays the original visualization with identified annotations, highlighted by a red bounding box when the user hovers over the corresponding issue or label. The right panel presents the analysis results, including detailed descriptions and labels of the detected misleading elements in the visualization.

Dataset. 30 target visualizations were selected. Among these, 24 were randomly selected from Lo and Qu [13]'s dataset. Six visualizations were additionally collected from online news media and included to the dataset by the authors. These visualizations represented misleading issues that Lo and Qu's prompted guidelines cannot capture. The visualizations were then grouped into 3 sets, S1, S2, and S3. Each set contained 10 visualizations, 8 from the previous dataset and 2 added by the authors. The grouping was conducted randomly.

Participants. Six participants were recruited from a local university. The participants were required *not* to be professional in visualization design. They were asked to self-report how confident they were with visualizations (Table 4). Then, the six participants were assigned to 3 groups (A, B, C), with 2 participants in each. Each group used our full system (*ours(ann)*), our system without visual annotations (*ours(text)*), and text results from Prompt #1 (*Lo*) with the three image sets in different orders according to a 3 × 3 latin square, as shown in Table 2. This design was chosen to minimize any learning effects that could arise from exposure to the visualizations or annotations.

Interface. Visualizations were given to the participants with an experiment interface. As Figure 2 shows, the interface displayed a single visualization on the left and its analysis result on the right. The *Lo* system displayed the results of Prompt #1 in markdown, with the final summary on top and results of previous prompts at the bottom. Our systems presented textual explanations from our pipeline. For the *ours(text + ann)* system, hovering interaction was enabled to introduce direct annotation on the visualization. When users hover their cursor on a certain issue or label, corresponding components are highlighted with red bounding boxes. In vice versa, when users hover their cursor on a bounding box, corresponding issues and labels are highlighted with background colors.

Procedure. Each participant analyzed 30 visualizations, divided into 10 visualizations for each system following between-subject design. The visualizations within each image set were presented in a randomized order. We asked participants (1) to examine each visualization and identify as many misleading elements as possible, referencing the analysis provided by the system. Then, they were asked (2) to sketch their proposed fixes for the issues identified in the first step and explain the reason. We gave participants a digital tablet displaying an identical version of the visualization to sketch on it. The identified issues, proposed fixes, and reasoning were collected for further analysis. To ensure consistency in engagement time across participants, a time limit of 3 minutes was imposed for each visualization. This process was repeated for all 30 visualizations. A Semi-structured interview about user experiences was conducted to gather feedback on the usability, effectiveness, and overall impressions.

Table 3: Quantitative results of the user study. Participants found the issues and fixes most accurately when they were provided with our system: full pipeline with *ours(text)* and *annotation*.

System	***Lo***	***ours(text)***	***ours(ann)***
Issues (#)	**137**	114	130
True Issue (%)	61.31	71.93	**73.85**
False Issue (%)	13.87	9.64	**3.84**
Partially True Issue (%)	1.46	0	0
Insignificant Issue (%)	23.36	**19.30**	22.31
Good Fix (%)	75.00	80.49	**85.42**
Incomplete Fix (%)	8.33	12.20	11.46
Bad Fix (%)	14.29	4.87	**2.08**
Unresolved (%)	4.76	2.44	**1.04**

Table 4: Demographics of the participants of the user study. The participants were asked to report how confident they were with visualizations from a 5-point Likert scale. Confidence denotes the degree to which participants are confident in their visualization expertise (1: novice, 5: expert).

ID	P1	P2	P3	P4	P5	P6
Age	23	23	22	24	24	31
Gender	M	M	M	M	M	F
Confidence (1–5)	4	4	4	4	1	2

5.2 Analysis

Both quantitative and qualitative analyses were conducted. From the collected data, each issue was tagged as *true*, *partially true*, *false*, or *insignificant*. *Partially true* issues were tagged when the interviewee failed to spot another issue that it completely relies on. *Insignificant* issues were tagged when they were technically true but had minimal impact as misleading. For example, it includes remarks on data credibility, temporal or social context, and usage of difficult vocabulary.

For *true* issues, their corresponding fixes were tagged as *good*, *incomplete*, *bad*, or *unresolved*. *Incomplete* fixes were tagged when the fix did improve on the issue but could not eliminate it. *Unresolved* was tagged when the participant only spotted an issue and could not come up with a solution. The first author initially performed the tagging, after which two additional authors joined to review the results. We conducted revision until the consensus of all three participating authors was met.

Qualitative analysis was applied to records from the user study. After participants used each system, they were asked about their experience with the system. This included what feature they used the most, what criteria they judged the visualizations upon, and how much they utilized their previous knowledge.

5.3 Findings

5.3.1 Quantitative Analysis

Table 3 shows an overview of the results of the quantitative analysis. **Effectiveness of our Pipeline in Revealing Misleading Designs.** *Ours(text)* helped participants detect fewer *false* and *insignificant* issues compared to the *Lo* system, reducing unnecessary cognitive effort during the verification. When annotations were added in *ours(ann)*, the number of *false* issues decreased further, while more *true* issues were identified. Consequently, the result revealed that both our pipeline and annotation designs effectively assist in detecting genuine errors.

Effectiveness of Annotations. The addition of visual annotations in *ours(ann)* improved participants performance. *Ours(ann)* achieved the lowest rate of *unresolved* and *bad* fixes, showing that annotations enhanced the explainability of visualization guidelines. By helping users pinpoint misleading elements, annotations not only increased the number of *true* issues but improved the quality of fixes. Compared to the *Lo* system, *ours(ann)* produced more good fixes and fewer bad fixes. Even against *ours(text)*, annotations led to better detection and correction, proved its effectiveness.

Trade-off between Accuracy and Quantity. The *Lo* system encouraged participants to identify a greater number of issues by providing extensive analysis and guidelines, including criteria where visualizations performed well. However, this exhaustive approach often resulted in the discovery of many insignificant or false issues and a lower rate of true issues overall. Conversely, the *ours(text)* and *ours(ann)* systems streamlined the process by highlighting only detected issues and their corresponding components, which guided participants toward more accurate and meaningful insights. This finding aligns with the results of our preliminary study in which guidelines tended to pick up too many insignificant issues. This can confuse users by overwhelming them with too much information.

5.3.2 Interview Results

Preference. The interview results imply that our system is most preferable for participants. Three out of the six interviewees chose the *ours(ann)* system as their favorite. P2 said, *"I liked how it gave me summarized results with bullet points."* P3 also noted that *"the system made it faster to find issues in the visualization."* P6 especially preferred the direct visual annotations. *"It was helpful to check how the visual annotations and analysis texts were related."*

Two participants chose the *Lo* system over the *ours(ann)* system. The main reason was that *"It gave the most information,"* as P1 mentioned. Still, P4 commented that *"the* `Lo` *system was most convenient for me, but I think that's because of the learning effect. Without that, the* `ours(ann)` *system might be easier to use."* P5 uniquely preferred the *ours(text)* system over the *ours(ann)* system. This was because *"without the annotations, it was clearest to see."*

The participants also pointed out the drawbacks of the systems. The *Lo* system tended to provide too much information, overwhelming the participants. P5 said, *"I'd rather study the graph than study the text."* For the *ours(ann)* system, there were differing opinions on the annotation scheme. P6 proposed, *"using effects like highlighter pens might be better than bounding boxes. The boxes sometimes cover up the visualization."*

Issues and Fixes. We find that participants show different patterns when fixing issues compared to when looking for them. While participants mostly relied on the provided analyses for identifying issues, they fixed them mainly relying on their own knowledge, or *'common sense'* as P3 denoted. When looking for issues, participants tended to consult the system. P5 said, *"The system helped a lot. It made it much faster to find issues in the visualizations."* However, when asked how they came up with improvement plans, P5 answered *"I think it's mostly pure intuition. I rarely read the system's recommendations in terms of fixing."*

Users tended to ignore suggestions when they believed the analysis was inaccurate. Participants with high confidence primarily utilized the system for verification purposes and were less influenced by its errors. However, both confident and less confident participants exhibited instances in which they mistakenly identified misleading issues in the visualization that were not actually present.

Expertise and Dependency. Participants who were more confident less depended on the systems provided. P6, who chose 2 from a 5-point likert scale when asked how confident they are with visualizations, said that *"I wouldn't have a clue if the system wasn't there. At first, all of the visualizations seemed completely normal to me."* In contrast, P2 said *"I mostly find all the issues by looking at the visualization directly. I used the system only to verify that I am right."* P2 answered 4 from the same 5-point likert scale question. Overall, participants who answered that they were familiar to visualizations tended to rely on their own knowledge when fixing the identified issues.

For identifying issues, participants with more expertise tended to focus on certain features which they thought were prominent. P1 and P2 both mentioned that they concentrated on the *Chart Interpretation* section while using the *Lo* system. Especially, P1 considered the *Cherrypicking* criteria most important from personal experience.

Annotation Effects. With visual annotations, participants could effectively determine whether a given issue is true by looking at the provided labels and corresponding annotations. P2 mentioned that *"When I look at the highlighted components and think that they don't match the explained issue, I can easily ignore it."* Also, participants completed given tasks faster with the *ours(ann)* system. P3 pointed out, *"Instead of going back and forth between the visualization and the analysis, I can directly see where in the image the problem sits."*

While participants who did not prefer the provided visual annotations reasoned that it was *"too simple"*, results show that this actually introduced the effect of reducing *false* issues and *bad* fixes. Participants tended to read through the long guidelines of the *Lo* system, and were easily persuaded by them. For example, when the *Lo* system claimed that the axis intervals were inconsistent, P3 said *"at first I thought they were normal, but since the system insists, they start to seem not quite right."* This visualization had no problems with its axes.

6 Discussion

6.1 Effectiveness of Visual Annotations

Our results show that integrating textual explanations with visual annotations is highly effective in terms of preventing users from misinterpreting visualizations. Participants were able to identify more real issues and establish more effective plans to fix the visualizations using our system. Also, while text-based approaches like the *Lo* system found the most issues, they also introduced a large number of *insignificant* issues. These long natural language analyses tended to overwhelm users, making them confused.

Meanwhile, our study implies that the effectiveness of visual annotations can be further explored. For example, multiple interviewees proposed that highlighter effects would aid in investigating the visualization compared to our current design (red bounding box). Such responses suggest that our current design may not be the most effective way to visually convey misleading concepts through annotations. Consequently, a systematic examination of various annotation designs would be a promising avenue for future work.

6.2 Limitations and Future Work

While our current pipeline is promising, several challenges remain. Processing time for complex visualizations limits real-time usability, and future efforts should aim to optimize the pipeline for scalability. In our pipeline, the segmentation stage takes an average of 83 seconds, with an extreme of 95 seconds depending on the environment and visualization. Utilizing more concrete segmentation models or specifically trained machine learning models may improve the pipeline, enabling direct querying from the user. Also, with more advanced segmentation models and multi-modal large language models, our pipeline can produce even more accurate and perceptive comments upon subtle or complex bias, triggered by interrelationships of multiple elements.

7 Conclusion

This study underscores the prevalence and impact of misleading visualizations, emphasizing the need for tools that enhance audience criticality. Our system addresses this challenge by combining segmentation models and LLMs to detect, annotate, and critique misleading elements in visualizations. Our evaluation demonstrates that users equipped with visual annotations on misleading visualization designs can more effectively escape from misinterpretations, fostering a more skeptical and informed approach to consuming visual data.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2023R1A2C200520911) and the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [NO.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)]. The ICT at Seoul National University provided research facilities for this study. We thank Sungbok Shin and Hyunwook Lee for their valuable feedback.

References

[1] J. Alexander, P. Nanda, K.-C. Yang, and A. Sarvghad. Can gpt-4 models detect misleading visualizations? In *2024 IEEE Visualization and Visual Analytics (VIS)*, pp. 106–110, 2024. doi: 10.1109/VIS55277.2024.00029 2, 3

[2] A. F. Blackwell et al. Cognitive dimensions of notations: Design tools for cognitive technology. In *Cognitive Technology: Instruments of Mind*, pp. 325–341. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001. 3

[3] Q. Chen, F. Sun, X. Xu, Z. Chen, J. Wang, and N. Cao. Vizlinter: A linter and fixer framework for data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):206–216, 2022. doi: 10.1109/TVCG.2021.3114804 2

[4] A. Diehl, A. Abdul-Rahman, M. El-Assady, B. Bach, D. Keim, and M. Chen. VisGuides: A Forum for Discussing Visualization Guidelines. In *EuroVis 2018 - Short Papers*, 2018. doi: 10.2312/eurovisshort.20181079 1

[5] A. Fan, Y. Ma, M. Mancenido, and R. Maciejewski. Annotating line charts for addressing deception. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, 2022. doi: 10.1145/3491102.3502138 1

[6] X. Fu, Y. Wang, H. Dong, W. Cui, and H. Zhang. Visualization assessment: A machine learning approach. In *2019 IEEE Visualization Conference (VIS)*, pp. 126–130, 2019. doi: 10.1109/VISUAL.2019.8933570 2

[7] A. K. Hopkins, M. Correll, and A. Satyanarayan. Visualint: Sketchy in situ annotations of chart construction errors. *Computer Graphics Forum*, 39(3):219–228, 2020. doi: 10.1111/cgf.13975 1, 2, 3

[8] D. Jung, W. Kim, H. Song, J.-i. Hwang, B. Lee, B. Kim, and J. Seo. Chartsense: Interactive data extraction from chart images. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, p. 6706–6717, 2017. doi: 10.1145/3025453.3025957 2

[9] A. Key, B. Howe, D. Perry, and C. Aragon. Vizdeck: self-organizing dashboards for visual analytics. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, p. 681–684, 2012. doi: 10.1145/2213836.2213931 2

[10] C. Lai, Z. Lin, R. Jiang, Y. Han, C. Liu, and X. Yuan. Automatic annotation synchronizing with textual description for visualization. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, p. 1–13, 2020. doi: 10.1145/3313831.3376443 2

[11] X. Lan and Y. Liu. "i came across a junk": Understanding design flaws of data visualization from the public's perspective. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):393–403, 2025. doi: 10.1109/TVCG.2024.3456341 2

[12] L. Y.-H. Lo, A. Gupta, K. Shigyo, A. Wu, E. Bertini, and H. Qu. Misinformed by visualization: What do we learn from misinformative visualizations? *Computer Graphics Forum*, 41(3):515–525, 2022. doi: 10.1111/cgf.14559 2

[13] L. Y.-H. Lo and H. Qu. How good (or bad) are llms at detecting misleading visualizations? *IEEE Transactions on Visualization and Computer Graphics*, 31(1):1116–1125, 2025. doi: 10.1109/TVCG.2024.3456333 2, 3, 4

[14] Y. Luo, X. Qin, N. Tang, and G. Li. Deepeye: Towards automatic data visualization. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pp. 101–112, 2018. doi: 10.1109/ICDE.2018.00019 2

[15] A. McNutt and G. Kindlmann. Linting for visualization: Towards a practical automated visualization guidance system. In *VisGuides: 2nd Workshop on the Creation, Curation, Critique and Conditioning of Principles and Guidelines in Visualization*, vol. 1, 2018. 1

[16] A. McNutt, G. Kindlmann, and M. Correll. Surfacing visualization mirages. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, p. 1–16, 2020. doi: 10.1145/3313831.3376420 2

[17] D. Moritz, C. Wang, G. L. Nelson, H. Lin, A. M. Smith, B. Howe, and J. Heer. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):438–448, 2019. doi: 10.1109/TVCG.2018.2865240 2

[18] V. T. Nguyen, K. Jung, and V. Gupta. Examining data visualization pitfalls in scientific publications. *Visual Computing for Industry, Biomedicine, and Art*, 4(1):27, Oct 2021. doi: 10.1186/s42492-021-00092-y 2

[19] A. V. Pandey, K. Rall, M. L. Satterthwaite, O. Nov, and E. Bertini. How deceptive are deceptive visualizations? an empirical analysis of common distortion techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, p. 1469–1478, 2015. doi: 10.1145/2702123.2702608 2

[20] J. Poco and J. Heer. Reverse-engineering visualizations: Recovering visual encodings from chart images. *Computer Graphics Forum*, 36(3):353–363, 2017. doi: 10.1111/cgf.13193 2

[21] M. D. Rahman, G. J. Quadri, B. Doppalapudi, D. A. Szafir, and P. Rosen. A qualitative analysis of common practices in annotations: A taxonomy and design space. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 3

[22] N. Ravi et al. Sam 2: Segment anything in images and videos, 2024. 3

[23] J. Rho and M. A. Rau. Exploring educational approaches to addressing misleading visualizations. *Educational Psychology Review*, 37(1):14, Feb 2025. doi: 10.1007/s10648-025-09988-0 2

[24] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, 2017. doi: 10.1109/TVCG.2016.2599030 2

[25] S. Shin, S. Hong, and N. Elmqvist. Visualizationary: Automating design feedback for visualization designers using llms, 2024. 2

[26] C. Stokes, C. X. Bearfield, and M. A. Hearst. The role of text in visualizations: How annotations shape perceptions of bias and influence predictions. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 3

[27] D. A. Szafir. The good, the bad, and the biased: five ways visualizations can mislead (and how to fix them). *Interactions*, 25(4):26–33, June 2018. doi: 10.1145/3231772 2

[28] A. Tarrell, A. Fruhling, R. Borgo, C. Forsell, G. Grinstein, and J. Scholtz. Toward visualization-specific heuristic evaluation. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, pp. 110–117, 2014. 1

[29] E. R. Tufte and P. R. Graves-Morris. *The visual display of quantitative information*, vol. 2. Graphics press Cheshire, CT, 1983. 1

[30] C. Xiong, C. Stokes, Y.-S. Kim, and S. Franconeri. Seeing what you believe or believing what you see? belief biases correlation estimation. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):493–503, 2023. doi: 10.1109/TVCG.2022.3209405 2

[31] C. Xiong, L. Van Weelden, and S. Franconeri. The curse of knowledge in visual data communication. *IEEE Transactions on Visualization and Computer Graphics*, 26(10):3051–3062, 2020. doi: 10.1109/TVCG.2019.2917689 2