Is this chart lying to me? Automating the detection of misleading visualizations

Jonathan Tonglet[*1,2,3], **Jan Zimny**[*1,2], **Tinne Tuytelaars**[2], **Iryna Gurevych**[1]
[1] Ubiquitous Knowledge Processing Lab (UKP Lab), Department of Computer Science, TU Darmstadt and National Research Center for Applied Cybersecurity ATHENE
[2] Department of Electrical Engineering, KU Leuven
[3] Department of Computer Science, KU Leuven
www.ukp.tu-darmstadt.de

Abstract

Misleading visualizations are a potent driver of misinformation on social media and the web. By violating chart design principles, they distort data and lead readers to draw inaccurate conclusions. Prior work has shown that both humans and multimodal large language models (MLLMs) are frequently deceived by such visualizations. Automatically detecting misleading visualizations and identifying the specific design rules they violate could help protect readers and reduce the spread of misinformation. However, the training and evaluation of AI models has been limited by the absence of large, diverse, and openly available datasets. In this work, we introduce Misviz, a benchmark of 2,604 real-world visualizations annotated with 12 types of misleaders. To support model training, we also create Misviz-synth, a synthetic dataset of 57,665 visualizations generated using Matplotlib and based on real-world data tables. We perform a comprehensive evaluation on both datasets using state-of-the-art MLLMs, rule-based systems, and image-axis classifiers. Our results reveal that the task remains highly challenging. We release Misviz, Misviz-synth, and the accompanying code.[1]

1 Introduction

Misleading visualizations are charts that distort the underlying data, typically by violating design principles, leading readers to draw inaccurate conclusions (Tufte and Graves-Morris, 1983; Pandey et al., 2015; Lauer and O'Brien, 2020; McNutt et al., 2020; Lo et al., 2022; Lisnic et al., 2023; Lan and Liu, 2025). While many arise from unintentional design errors, misleading visualizations are also deliberately crafted by malicious actors to spread disinformation and manipulate public understanding, especially during crises such as the COVID-19 pandemic, where misleading charts circulated widely on social media (Correll and Heer, 2017; Lisnic et al., 2023; Tartaglione and de Wit, 2025). Prior work has shown that both humans (Pandey et al., 2014, 2015; O'Brien and Lauer, 2018; Yang et al., 2021; Ge et al., 2023; Rho et al., 2023) and MLLMs (Bendeck and Stasko, 2025; Chen et al., 2025; Pandey and Ottley, 2025; Tonglet et al., 2025) are easily deceived by such visualizations in question-answering tasks.

The deceptive features in these charts, or *misleaders* (Lisnic et al., 2023), often reside in subtle details easily missed by readers, such as axis tick intervals. Furthermore, misleaders are highly diverse: the latest taxonomies identify over 70 distinct types spanning a wide range of chart types, including bar charts, pie charts, and choropleth maps (Lo et al., 2022; Lan and Liu, 2025). In some cases, multiple misleaders affect the same visualization (Lo et al., 2022). Figure 1 shows 12 real-world examples of misleading visualizations.

Automatically classifying whether a visualization is misleading and identifying which misleaders affect it, if any, can enable timely warnings to chart designers and readers and help prevent the spread of misinformation. This task is framed as a multi-label classification problem. While early work relied on rule-based systems called linters (Hopkins et al., 2020; Fan et al., 2022), recent studies have explored the use of MLLMs (Lo and Qu, 2025; Alexander et al., 2024). However, these approaches were evaluated on distinct datasets, which are either small or closed-access, limiting comparability and hindering progress.

In this work, we introduce Misviz, a large, diverse, and open benchmark comprising 2,604 real-world visualizations spanning 12 types of misleaders. It reflects scenarios in which detection models could flag visualizations published on the web. In Misviz, 70% of the visualizations contain up to three misleaders, while the remaining 30% are

[*] These authors contributed equally to this work.
[1] github.com/UKPLab/acl2026-misviz

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8823–8844
July 2-7, 2026

Figure 1: Examples of the 12 types of misleaders included in Misviz. Appendix A explains how these visualizations misrepresent their underlying data table.

non-misleading. To support model training, we also release Misviz-synth, a synthetic dataset generated with Matplotlib (Hunter, 2007) using real-world data tables. Misviz-synth reflects scenarios in which detection models assist chart designers in identifying misleaders unintentionally introduced into their charts. The dataset includes not only the visualizations but also their underlying data tables, Python code snippets, and axis metadata, enabling the training of chart de-rendering models.

We conduct extensive experiments with three approaches: (a) state-of-the-art MLLMs, (b) a new rule-based linter that inspects axis metadata for design rule violations, and (c) new classifiers that take the visualization alone or in combination with the axis metadata as input. For (b) and (c), we fine-tune DePlot (Liu et al., 2023) to extract axis metadata as an intermediate step. Our experiments address the following research questions (RQs). **RQ1**: Which type of model performs best on real-world or synthetic instances? **RQ2**: Can detection models trained on synthetic instances generalize to real-world cases? **RQ3**: Can axis extraction models trained on synthetic instances generalize to real-world cases?

Our results show that MLLMs perform best on real-world visualizations, while linters and image-axis classifiers outperform them on synthetic ones, benefiting from the availability of training data for both axis extraction and misleader detection. While the fine-tuned DePlot can extract axes from Misviz-synth, it does not generalize well to Misviz, reducing the performance of the linter and classifier.

In summary, our contributions are as follows: (1) We introduce Misviz and Misviz-synth, the first large-scale open datasets for misleading visualization detection. (2) We propose a new linter and a new classification method that combines image and extracted axis metadata as input. (3) We conduct a comprehensive evaluation and error analysis, highlighting the strengths and weaknesses of each method and identifying directions for future work.

2 Related work

The first attempts to detect misleading visualizations relied on rule-based systems called linters (McNutt and Kindlmann, 2018; Hopkins et al., 2020; Chen et al., 2022). These linters assume the availability of the underlying data table or the chart code, which restricts their applicability to real-world scenarios. Fan et al. (2022) and Biselli et al. (2025) overcome these limitations by extracting tables using OCR tools from real-world visualizations before applying rule checks. However, the accuracy of these real-world linters depends heavily on the quality of the intermediate OCR step, which can vary widely (Biselli et al., 2025). Real-world linters have previously been evaluated in small-scale user studies with human-in-the-loop correction of OCR errors.

Others have explored the potential of MLLMs for the task. Lo and Qu (2025) evaluated four MLLMs with different prompts on a dataset of 150 real-world visualizations, sourced from the corpus of Lo et al. (2022) for misleading cases. They found that detection accuracy decreased as more misleader types were included in the prompt. Alexander et al. (2024) focused on GPT-4 (OpenAI, 2023), using visualizations from the social media platform X (Lisnic et al., 2023). However, access to this dataset requires a paid API, and reproducibility is further hindered by the platform's frequent removal of posts. In a parallel work, Das and Mueller (2026) proposed a prompt with which SOTA MLLMs achieve high accuracy on a subset of the corpus of Lo et al. (2022).

Recently, Maciborski et al. (2025) fine-tuned a convolutional neural network for the task, achieving high accuracy on synthetic instances.

Table 1 compares prior datasets with Misviz and Misviz-synth. Misviz is over fifteen times larger than the dataset of Lo and Qu (2025). Unlike Alexander et al. (2024), it does not rely on paid APIs for data collection and ensures long-term access to all instances by archiving them on the Wayback Machine.[2] Misviz-synth is two to seven times larger than other synthetic datasets, and includes several more misleaders and chart types (Arif et al., 2024; Maciborski et al., 2025). In contrast to other synthetic datasets, Misviz-synth provides the underlying table, code, and axis metadata. The latter is necessary to fine-tune DePlot for axis extraction and answer our research questions.

[2] web.archive.org

3 Misviz

3.1 Selected misleaders

Misviz covers 12 types of misleaders, selected from the 74 categories defined in the taxonomy of Lo et al. (2022), based on four key criteria. First, we excluded misleaders that are rarely observed in real-world scenarios. To determine this, we used misleader frequency statistics from the corpus of Lo et al. (2022) and discarded all categories with fewer than 15 instances. Second, we removed reasoning misleaders, i.e., misleaders that do not directly break chart design rules and are deceiving only in the context of a specific claim (Lisnic et al., 2023). Third, we remove misleaders which confuse rather than deceive. As noted by Lo et al. (2022), the taxonomy includes both misleaders that distort the underlying data, the focus of this work, and others that may hinder readability or clarity without altering the interpretation of the data, such as *missing titles* or *overplotting*. Fourth, we excluded misleaders that require specific domain knowledge to be identified. For example, using red to represent Democrats and blue to represent Republicans in a chart violates *color conventions*, but detecting this misleader requires familiarity with U.S. politics. Such misleaders require domain expertise that is beyond the reach of crowdworkers.

We define each selected misleader briefly below (Lo et al., 2022; Ge et al., 2023; Lan and Liu, 2025). They cover together 62.3% of all instances from the real-world corpus of Lo et al. (2022). Each of them is represented with an example in Figure 1 and in Appendix A.

Misrepresentation: the value labels displayed do not match the sizes of their visual encodings; e.g., bars may be drawn disproportionately to their corresponding numerical values.

3D: the visualization includes 3D effects, distorting the size of visual encodings.

Truncated axis: an axis does not start from zero, thus exaggerating differences between values.

Inappropriate use of pie chart: a pie chart does not display data in a part-to-whole relationship.

Inconsistent binning size: a variable, such as years or ages, is grouped in unevenly sized bins.

Discretized continuous variable: a continuous variable is cut into discrete categories, thus exaggerating the difference between boundary cases.

Inconsistent tick intervals: the ticks in one axis are evenly spaced, but their values are not, e.g., the tick values sequence is 10, 20, 40, 45.

Dataset	Instances	Misleader types	Chart types	% non-misleading	Open access	Real-world	Multi-label	Axes, table, code
MISCHA-QA (Arif et al., 2024)	8,201	4	3	49	✓	✗	✗	✗
DCDM (Maciborski et al., 2025)	24,480	5	3	51	✓	✗	✗	✗
Alexander et al. (2024) - *design misleaders*	1,460	7	> 5	50	✗	✓	✗	✗
Lo and Qu (2025)	150	21	> 5	16	✓	✓	✗	✗
Misvisfix (Das and Mueller, 2026)	450	74	> 5	20	✓	✓	✓	✗
Misviz-synth (ours)	57,665	12	5	39	✓	✗	✗	✓
Misviz (ours)	2,604	12	> 5	31	✓	✓	✓	✗

Table 1: Existing datasets for misleading visualization detection.

Dual axis: there are two independent and parallel numerical axes with different scales.

Inappropriate use of line chart: a line chart is used in unusual ways, e.g., with categorical data.

Inappropriate item order: the tick labels of an axis are sorted in an unconventional way, e.g., dates are not shown chronologically.

Inverted axis: an axis is displayed in a direction opposite to conventions.

Inappropriate axis range: the axis range is either too broad or too narrow, minimizing or exaggerating the real trend.

3.2 Data collection

We obtain visualizations from three sources.

(a) We collect instances from the corpus of Lo et al. (2022) that contain at least one of the 12 selected misleaders. We apply perceptual hashing to remove duplicates. We then manually discard instances with low resolution, those that display the name of the misleader, or those that show both a misleading chart and a corrected version side by side. Appendix B shows removed examples.

(b) We use the misleading visualizations from the website WTF Visualizations,[3] previously annotated by Lan and Liu (2025) for taxonomy construction. We align their misleader categories with those of Lo et al. (2022), as explained in Appendix C, retain only those matching one of our 12 target misleaders, and remove duplicates.

(c) We access a large collection of unlabeled visualizations from *r/dataisugly* and *r/dataisbeautiful*.[4] The former is an online community focused on sharing misleading visualizations, while the latter features high-quality examples and serves as our source of non-misleading instances. We begin by removing duplicates, then hire annotators to assign labels, as explained in Section 3.3.

[3] viz.wtf

[4] kaggle.com/datasets/bcruise/reddit-data-is-beautiful-and-ugly

3.3 Data labeling

We split the labeling of visualizations from *r/dataisugly* and *r/dataisbeautiful* into three annotation tasks. For each task, we hired three crowdworkers from Prolific and paid them £9 per hour. To ensure sufficient familiarity with visualizations, we required annotators to hold a PhD degree. Inter-annotator agreement (IAA) was evaluated using Fleiss' κ (Fleiss, 1971) on an overlapping set of 30 instances across all crowdworkers. Appendix D shows the annotation interface and instructions.

In the first task, crowdworkers assigned one or more chart types to each visualization. Five categories were available: bar charts, line charts, maps, scatterplots, and others. The "others" category includes less frequent chart types such as treemaps. The crowdworkers achieved a high IAA (0.71).

In the second task, crowdworkers labeled visualizations from *r/dataisugly* with zero to three misleaders. They received detailed guidelines with definitions and examples for each misleader. Some visualizations were assigned no label and removed from the corpus, typically because they contained a misleader outside of our selected set. A moderate IAA of 0.53 is achieved. This is a reasonable outcome, given the task's complexity with 12 categories and multiple possible combinations.

In the final task, crowdworkers verified that visualizations from *r/dataisbeautiful* were not misleading. Visualizations identified as containing one or more misleaders were removed from the corpus. The Fleiss' κ was 0.78, reflecting high IAA.

3.4 Bounding box labeling

We recruited ten PhD students to annotate misleading visualizations using the VIA tool (Dutta et al., 2016; Dutta and Zisserman, 2019). Annotators drew bounding boxes around relevant misleader features, e.g., the initial tick mark on a truncated axis. Three misleaders were excluded: *misrepresentation*, *3D*, and *inappropriate use of pie chart*, because bounding boxes are not suitable for representing them. Each student annotated up to 95

instances. We calculated IAA on a shared subset of 27 instances, three per misleader. Using an Intersection over Union (IoU) threshold of 0.4, the annotators achieved an IAA of 0.81, indicating strong agreement.

3.5 Data statistics

Misviz is divided into a few-shot development set (5%), a validation set for hyperparameter and prompt tuning (15%), and a held-out test set (80%). 78% of the visualizations contain one of the three main chart types: bar chart, line chart, or pie chart. While 94% of the visualizations contain a single chart type, 5 and 1% contain two or three different chart types, respectively. The split is stratified to ensure a balanced distribution of misleaders and chart types. Among the misleading visualizations, 85, 14, and 1% contain one, two, or three misleaders, respectively. The most frequent misleader is misrepresentation, present in 32% of the visualizations. This aligns with prior findings (Lo et al., 2022; Lan and Liu, 2025). The next most frequent categories are 3D effects and truncated axes, appearing in 14 and 9% of the visualizations, respectively. All other misleaders occur in 1 to 5% of the instances. Appendix E provides detailed distributions of misleaders and chart types.

Although we do not provide language distributions, it is worth noting that the visualizations span many languages. The types of images also vary significantly. While most are screenshots, some are pictures of visualizations printed on paper or displayed on screens. Appendix F provides cases of non-English and non-screenshot visualizations.

We did not leverage the large unlabeled corpus collected by Lo et al. (2022). However, this resource could be used in future work to scale the dataset, as discussed in Appendix G.

4 Misviz-synth

4.1 Data collection

We collect real-world data tables from Our World in Data, an open-access data platform that covers a wide range of domains, including health, education, and the economy.[5]

4.2 Synthetic visualizations generation

We design a two-step rule-based system to generate synthetic visualizations, illustrated in Figure 2.

[5] ourworldindata.org

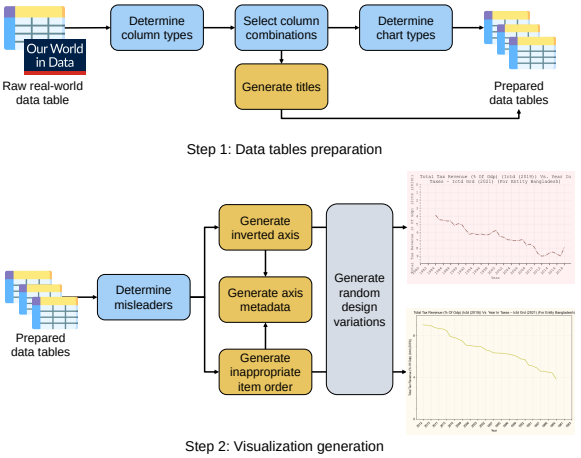


Figure 2: The two-step process to create the synthetic visualizations of Misviz-synth based on real-world data.

In the first step, given a raw real-world data table, we assign one or more types to each column. There are 10 different column types, explained in Appendix H. Based on these types, we select valid column combinations that require at least one numerical column and a natural key for each row. This natural key can consist of one or more categorical or temporal columns. If the natural key involves multiple columns, we fix all but one to a constant value. For instance, if the numerical column is *Win ratio* and the natural key includes *Football club* and *Year*, we set *Year* to 2021 to compare the win ratios of football clubs for that year. Chart titles are automatically generated using templates based on the selected columns. We then determine which chart types, among bar, line, and pie charts, are suitable for the data. For example, line charts are only compatible with temporal data, not categorical data. Each valid configuration produces a "prepared" data table.

In the second step, we generate visualizations using hand-crafted plotting functions for each chart type–misleader pair. For each prepared data table, we identify the misleaders applicable given the column types and selected chart types. One or more misleading visualizations are generated, each with exactly one misleader from the applicable set. Additionally, non-misleading visualizations are generated for a random subset of the prepared tables.

For bar, line, and scatter plots, we automatically extract axis metadata. Since pie charts and maps lack axes, they do not have axis metadata. Axis metadata is stored as a table with four columns. (1) *Seq* indicates the sequential order of the tick marks along an axis. Horizontal axes are read from left to

right, while vertical axes are read from bottom to top. (2) *Axis* indicates the name of the axis, e.g., x, y1, or y2, in case of dual axis. (3) *Label* is the tick label. (4) *Relative position* is a normalized float indicating the tick mark's position, with spacing expressed relative to the distance between the first two tick marks. Examples of axis metadata are provided in Appendix I.

To increase visual diversity, we apply random variations in font size, background color, and axis-label positions. The complete list of variations is given in Appendix J.

Each instance is saved along with its title, data table, Matplotlib code, and axis metadata.

4.3 Data statistics

Misviz-synth contains 57,665 visualizations, distributed into stratified train-large (80.1%), train-small (9.6%), dev (3.1%), validation (3.2%), and test sets (4.1%). Misviz-synth covers the five most common chart types in Misviz: bar, line, and pie charts; scatterplots; and maps. Unlike Misviz, each visualization in Misviz-synth contains at most one chart type and one misleader, making it slightly less diverse than Misviz.

5 Experiments

5.1 Baselines

We consider three categories of baseline models, illustrated in Figure 3.

Zero-shot MLLMs We evaluate the zero-shot capabilities on one run of the following commercial and open-weight MLLMs: GPT-4.1 and GPT-o3 (OpenAI, 2023), Gemini-2.5-Flash-Lite (Gemini-Team, 2024), Qwen-2.5-VL in 7B, 32B, and 72B variants (Qwen-Team, 2025), and InternVL3 in 8B, 38B, and 78B variants (Zhu et al., 2025). These models were selected for their strong performance on the ChartQA benchmark (Masry et al., 2022). The prompt includes the task description and definitions of the misleaders. The prompt is provided in Appendix K.

Rule-based linter We design a linter that detects misleaders by applying rule-based checks to the axis metadata of a visualization. Each misleader corresponds to a specific rule. If a visualization passes all checks, it is classified as having no misleaders. We rely on axis metadata rather than the underlying data table because it enables detection of a broader range of misleaders: truncated axis, inverted axis, dual axis, inconsistent tick intervals, inconsistent binning size, and inappropriate item order. Only the latter two misleaders could also be detected from the data table alone. Detailed descriptions of all rule checks are provided in Appendix L. Misleaders that require visual interpretation or contextual knowledge, such as misrepresentation or inappropriate axis ranges, are not covered by the linter. We evaluate its performance using both ground truth and predicted axis metadata.

Image-axis classifiers We train two classifiers: one that takes only the visualization image as input, and another that combines the image with its axis metadata. Visualizations are encoded using a frozen image encoder, while axis metadata is embedded using a frozen table encoder. We use the image encoder of TinyChart, a specialized chart understanding MLLM (Zhang et al., 2024), while the table encoder is TaPas (Herzig et al., 2020). For the second classifier, the resulting image embeddings are concatenated with the [CLS] token from the axis metadata embeddings. The image embedding or concatenated embeddings are passed through a trained classification head. The classifiers are trained on Misviz-synth to predict either one misleader per visualization or none.

Axis extraction The linter and the second classifier require axis metadata as input. While ground truth metadata is available for Misviz-synth, this is not the case for Misviz or real-world scenarios, where only the visualization image is accessible. To address this, we implement an intermediate axis extraction step using DePlot (Liu et al., 2023). Since DePlot was originally only trained for chart-to-table extraction, we fine-tune it on (image, axis metadata) pairs from Misviz-synth.

5.2 Evaluation metrics

We evaluate model performance using six metrics. The first four assess the binary classification of visualizations as misleading or not: Accuracy (Acc), Precision (Pre), and Recall (Rec) for the misleading class, and macro-F1 score (F1). The remaining two metrics are computed on the subset of visualizations that are misleading and evaluate the correct identification of specific misleaders. Exact Match (EM) assigns a score of 1 if the set of predicted misleaders exactly matches the ground truth. Partial Match (PM) assigns a score of 1 if the predicted misleaders are a subset of the ground truth.

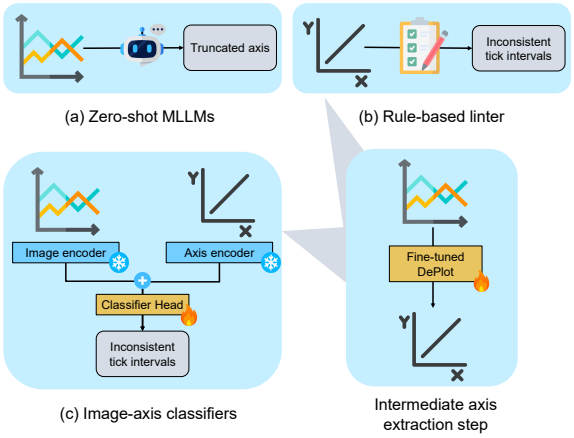


Figure 3: The three types of baselines included in the experiments. The linter and one classifier require axis extraction as an intermediate step.

5.3 Implementation details

Open-weight models are accessed via Transformers (Wolf et al., 2020), while commercial models are accessed via their official APIs. For all MLLMs, we set the temperature to 0, except for GPT-o3, which uses the default value of 1. Results for the image-axis classifiers are reported as averages over three different seeds (123, 456, 789), with standard deviations reported in Appendix M. Additional information and hyperparameter values are provided in Appendix N.

5.4 Results

Table 2 compares the performance of all models on the test sets of Misviz-synth and Misviz.

We organize our insights around the following RQs. **RQ1**: Which type of model performs best on real-world or synthetic instances? **RQ2**: Can detection models trained on synthetic instances generalize to real-world cases? **RQ3**: Can axis extraction models trained on synthetic instances generalize to real-world cases?

There are substantial performance differences across MLLMs, with GPT models achieving the best scores on both datasets, in particular for EM and PM. We attribute this to their strong OCR capabilities. For the Qwen2.5-VL family, performance improves with model scale. In particular, the largest version, 72B, is required to achieve EM scores above 11%. For the InternVL3 family, the best model size depends on the dataset: 38B for Misviz, 78B for Misviz-synth.

MLLMs outperform linters and image-axis classifiers on the real-world visualizations of Misviz. This trend does not hold for Misviz-synth, where MLLMs are outperformed in both F1 and EM. We attribute this to two main factors. First, Misviz-synth instances often contain many axis ticks, making extracted axis metadata a particularly informative feature for misleader detection. Second, the axis extractor and image-axis classifiers directly benefit from being trained on the Misviz-synth train set, resulting in stronger in-domain performance.

> **Insight 1.** MLLMs perform better on Misviz, while the image-axis classifiers and linters are the best on Misviz-synth (RQ1).

The linter with ground-truth axis metadata achieves high precision on Misviz-synth, demonstrating the usefulness of axis metadata for misleader detection and the precision guarantees offered by a rule-based system. However, the linter only covers six misleaders, negatively affecting recall and EM. Using predicted axis metadata rather than ground truth metadata has only a small negative effect on all metrics for Misviz-synth, with F1 and EM scores remaining above those of the best MLLMs. This indicates that the fine-tuned DePlot model reliably extracts axis information from Matplotlib-generated charts. However, on Misviz, the EM and PM are very low, dropping by around 40 points compared to Misviz-synth. This shows that DePlot fails to generalize to real-world visualizations because the diversity of axis designs in the Misviz-synth training instances is lower.

> **Insight 2.** The linter has high precision for a subset of misleaders, but is very sensitive to axis extraction errors (RQs 1 and 3).

> **Insight 3.** The axis extractor fine-tuned on synthetic visualizations has limited generalizability to real-world ones (RQ 3).

On Misviz-synth, the image-axis classifiers achieve higher EM and PM than all other baselines. The classifier with axis metadata is the strongest, further highlighting the usefulness of that input modality. Their scores for binary classification on Misviz decrease only slightly compared to Misviz-synth. As a result, the classifier with axis metadata outperforms all open-weight MLLMs except Qwen2.5-VL-72B. However, the EM and PM

	Misviz						Misviz-synth					
	Acc	Pre	Rec	F1	EM	PM	Acc	Pre	Rec	F1	EM	PM
Zero-shot MLLMs												
Qwen2.5-VL-7B	41.9	66.1	33.5	41.8	5.2	9.0	54.5	68.7	52.9	52.8	10.8	10.8
Qwen2.5-VL-32B	73.7	73.9	95.9	59.4	4.8	6.2	65.3	66.8	93.5	48.7	3.7	3.7
Qwen2.5-VL-72B	72.3	76.6	86.5	64.1	26.5	35.0	67.2	68.5	89.9	57.8	29.0	29.0
InternVL3-8B	63.1	68.5	86.8	45.1	22.3	28.9	63.1	65.4	82.7	44.2	10.2	10.2
InternVL3-38B	59.2	76.5	59.4	56.8	28.5	37.3	58.3	73.0	57.6	57.1	27.9	27.9
InternVL3-78B	55.6	68.1	67.8	48.0	23.8	30.9	63.0	65.3	89.2	50.1	33.2	33.2
Gemini-2.5-Flash-Lite	54.0	63.3	68.4	44.7	29.7	39.4	64.6	65.4	**94.3**	48.7	22.0	22.0
GPT-4.1	**84.1**	84.5	**94.3**	79.6	53.6	64.0	67.1	72.7	77.4	63.4	42.7	42.7
GPT-o3	83.5	**86.6**	90.1	**80.0**	**58.8**	**67.5**	68.2	78.4	69.2	66.9	44.3	44.3
Rule-based linter												
Linter w. axis (ground truth)	-	-	-	-	-	-	69.4	99.7	52.2	69.4	51.4	51.4
Linter w. axis (DePlot)	36.5	63.1	20.4	36.1	6.6	7.8	67.9	**98.7**	50.3	**67.9**	47.6	47.6
Image-axis classifiers												
Image	70.1	73.7	88.3	58.7	11.1	14.9	72.0	71.9	92.2	64.7	68.0	68.0
Image w. axis (DePlot)	72.8	74.6	92.0	60.9	12.3	17.1	**72.5**	72.1	92.6	65.2	**69.5**	**69.5**

Table 2: Performance (%) on the test sets of Misviz and Misviz-synth. The best results, excluding the linter with ground truth axis metadata, are marked in **bold**.

scores drop by more than 50 points from Misviz-synth to Misviz. This implies that the classifiers cannot generalize their ability to make fine-grained predictions of misleader categories from synthetic to real-world visualizations.

Insight 4. Classifiers trained on Misviz-synth can generalize to Misviz for binary classification (RQ 2).

The best EM scores remain low on both datasets, underscoring the difficulty of identifying which misleaders affect a visualization. For binary classification, most baselines tend to favor recall over precision, with the notable exception of the linter. Future work could aim for a better balance to improve overall F1 performance.

Based on these observations, rule-based linters and image-axis classifiers are best suited for controlled environments where table and axis metadata are available. They can assist chart designers by automatically detecting misleaders introduced unintentionally. In contrast, for flagging misleading visualizations encountered online, where only the image is available, MLLMs are the better option.

Appendix O reports the image-axis classifiers and linter results on Misviz per number of chart type and per number of misleader, providing deeper insights into the generalization gap. Appendix P provides a prompt sensitivity analysis using the best open-weight models.

	Misviz	Misviz-synth
Incorrect use of misleader definition	14	13
Incorrect measurement of object size	10	3
Incorrect extraction of axis metadata	2	14
Complex chart type	2	0
No detection of 3D effects	2	0

Table 3: Manual analysis of 30 prediction errors with GPT-4.1 on Misviz and Misviz-synth test sets.

5.5 Manual error analysis

We manually analyze two random samples of 30 incorrect predictions by GPT-4.1. Errors are categorized into different types, as reported in Table 3, with examples in Appendix Q.

Around half the errors in both datasets are due to incorrect uses of misleader definitions. For instance, several instances are incorrectly classified as *dual axis* because they display multiple lines. However, this misleader applies only when there are two distinct vertical axes. On Misviz, several instances are misclassified as *inappropriate use of pie chart* because the values shown do not sum to 100. GPT-4.1 fails to see that the pie slice labels are absolute values rather than percentages.

Many errors in Misviz are due to incorrect measurement of the sizes of visual encodings, such as pie slices and bar heights, which are critical for detecting *misrepresentation* by comparing the measured sizes with the value labels.

In Misviz-synth, many errors arise from incomplete or incorrect parsing of axis metadata. For example, GPT-4.1 frequently overlooks *inconsistent tick intervals* on long temporal or numerical

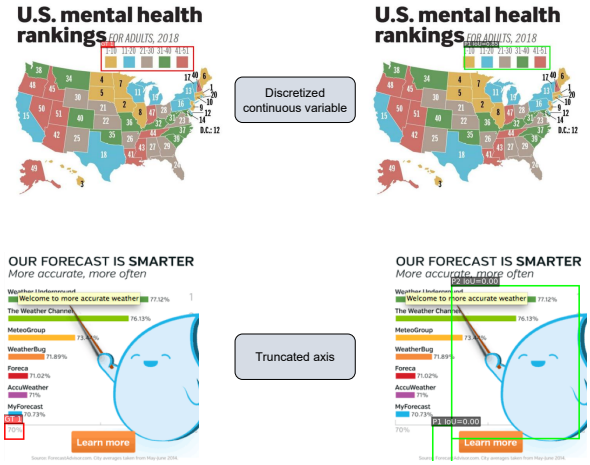


Figure 4: Examples of bounding box predictions for two instances of Misviz. The ground truth boxes are shown on the left, and the predictions on the right.

axes with numerous tick marks.

Two additional error categories are unique to Misviz. Some real-world visualizations are complex in their structure, sometimes containing more than four charts, which overwhelms the model with excessive visual information. In other cases, GPT-4.1 fails to detect *3D* effects, the only misleader that does not require analyzing any displayed labels.

5.6 Bounding box generation

We conduct a small-scale experiment to assess the ability of MLLMs to localize misleaders in visualizations using bounding boxes. Such localized predictions can serve as a complementary form of explanation for end users, alongside the predicted misleader labels. We evaluate Qwen2.5-VL-72B, the strongest open-weight MLLM, on the Misviz val set. The prompt is provided in Appendix K. We use IoU as a metric, matching each predicted bounding box to its nearest ground-truth. The MLLM achieves an IoU of 13.1%, a low score indicating that localization is challenging. Figure 4 shows two examples. In the top example, the MLLM accurately predicts a bounding box around the legend, indicating the discretized scale. In the bottom example, the MLLM fails to select the starting tick on the horizontal axis, indicating truncation.

6 Conclusion

In this work, we introduce two datasets for misleading visualization detection, covering 12 types of misleaders. Misviz is a real-world benchmark. Misviz-synth is a synthetic dataset generated from real-world tables, suitable for both training and evaluation. We propose a new linter and a trained classifier that leverage the axis metadata extracted from a visualization by a fine-tuned DePlot model. We conduct a comprehensive evaluation with three types of models: zero-shot MLLMs, linters, and image-axis classifiers. Our results show that the task poses several challenges. MLLMs perform best on real-world visualizations, while image-axis classifiers have an edge on synthetic ones.

Limitations

We identify three main limitations in this work.

First, Misviz-synth is less diverse than its real-world counterpart, Misviz. It does not include visualizations with multiple charts or multiple misleaders. Additionally, using the Matplotlib library imposes constraints; for instance, it does not support 3D pie charts. Future work should explore alternative plotting libraries to broaden coverage across chart types and misleader categories, thereby improving generalization to Misviz.

Second, the set of 12 misleader categories addressed in this work represents only a subset of those found in practice. While our selection was carefully made based on three criteria, future efforts should expand the taxonomy to include more misleaders. Notably, it would be valuable to incorporate misleaders that require domain knowledge to detect, such as *violating color conventions*. Furthermore, our selection focuses on design misleaders. However, there are also reasoning misleaders in which no design rules are broken. Instead, they mislead through incomplete and dubious data (Lo et al., 2022), or deceptive titles and annotations (Lisnic et al., 2023; Alexander et al., 2024). Such reasoning misleaders would require very different detection approaches and deserve more focus in future work.

Third, the boundaries between some misleader categories are not always clear-cut. For instance, *inappropriate item order* and *inverted axis* could be considered equivalent when a temporal axis is shown in reverse chronological order. Most cases of *inappropriate item order* with maps are usually also cases of *discretized continuous variables*, although only the former appears as the dominant misleader label in Misviz-synth. These label ambiguities are not captured by the current EM metric, which may slightly underestimate models' true performance in detecting misleading visualizations.

Ethics statement

Social impact Misleading visualizations are frequently used by malicious actors to spread misinformation online, particularly on social media platforms (Correll and Heer, 2017; Lisnic et al., 2023). They may also result from unintentional design choices by chart designers. This work contributes both a dataset and new baselines to help mitigate the negative impact of misleading visualizations by enabling their automatic detection. The Misviz and Misviz-synth datasets are intended solely for academic research. As with other resources developed for misinformation detection, there is a potential risk of dual use, where malicious actors exploit detection models in adversarial settings to craft misleading content that evades detection. However, we believe that the potential benefits of this research, such as assisting chart designers and protecting readers, outweigh these risks.

Misinformation content Misviz includes real-world disinformation examples. To preserve the authenticity and diversity of misleading visualizations encountered in the wild, we did not filter or censor such content.

Dataset access Our code and dataset annotations are released under the Apache 2.0 and CC BY-SA 4.0 licenses, respectively. We do not hold the rights to the visualization images in Misviz. Therefore, we do not distribute them directly. Instead, we provide image URLs in the Misviz dataset file, along with a script to download them. To ensure long-term accessibility, we have verified that all images are available on the WaybackMachine and included the corresponding archive URLS in the dataset file.

AI assistants use AI assistants were used in this work to assist with writing by correcting grammar mistakes and typos.

Acknowledgments

This work has been funded by the LOEWE initiative (Hesse, Germany) within the emergenCITY center (Grant Number: LOEWE/1/12/519/03/05.001(0016)/72), by the German Federal Ministry of Research, Technology and Space and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE, and by the Flanders AI Research Program. Figures 2, 3, 16, and 17 have been designed using resources from Flaticon.com. We thank Liesbeth Allein, Luke Bates, Manisha Venkat, Max Glockner, and Vivek Gupta for our insightful discussions on misleading visualizations. We are grateful to Niklas Traser for preparing the annotation interface for crowdworkers and implementing parts of the code for generating Misviz-synth. We also thank Shivam Sharma, Shivam Sharma (Junior), and Germán Ortiz for their feedback on an early draft of this work.

References

Jason Alexander, Priyal Nanda, Kai-Cheng Yang, and Ali Sarvghad. 2024. Can gpt-4 models detect misleading visualizations? In *2024 IEEE Visualization and Visual Analytics (VIS)*, pages 106–110.

Syraj Arif, Jane Swingler, and Andrew Wihardja. 2024. Mischa-qa.

Alexander Bendeck and John Stasko. 2025. An empirical evaluation of the gpt-4 multimodal language model on visualization literacy tasks. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):1105–1115.

Tom Biselli, Katrin Hartwig, Niklas Kneissl, Louis Pouliot, and Christian Reuter. 2025. Chartchecker: A user-centred approach to support the understanding of misleading charts. In *Proceedings of the 2025 ACM Designing Interactive Systems Conference*, DIS '25, page 2075–2102, New York, NY, USA. Association for Computing Machinery.

Qing Chen, Fuling Sun, Xinyue Xu, Zui Chen, Jiazhe Wang, and Nan Cao. 2022. Vizlinter: A linter and fixer framework for data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):206–216.

Zixin Chen, Sicheng Song, KaShun Shum, Yanna Lin, Rui Sheng, Weiqi Wang, and Huamin Qu. 2025. Unmasking deceptive visuals: Benchmarking multimodal large language models on misleading chart question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 13756–13789, Suzhou, China. Association for Computational Linguistics.

Michael Correll and Jeffrey Heer. 2017. Black hat visualization. In *Workshop on Dealing with Cognitive Biases in Visualisations (DECISIVe), IEEE VIS*.

Amit Kumar Das and Klaus Mueller. 2026. MisVisFix: An Interactive Dashboard for Detecting, Explaining, and Correcting Misleading Visualizations using Large Language Models . *IEEE Transactions on Visualization & Computer Graphics*, 32(01):134–144.

A. Dutta, A. Gupta, and A. Zissermann. 2016. Vgg image annotator (via). Version: 2.0.12, Accessed: 15-10-2026.

Abhishek Dutta and Andrew Zisserman. 2019. The via annotation software for images, audio and video. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM 19, New York, NY, USA. ACM.

Arlen Fan, Yuxin Ma, Michelle Mancenido, and Ross Maciejewski. 2022. Annotating line charts for addressing deception. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378–382.

Lily W. Ge, Yuan Cui, and Matthew Kay. 2023. Calvi: Critical thinking assessment for literacy in visualizations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Gemini-Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. Technical report, Google.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Aspen K. Hopkins, Michael Correll, and Arvind Satyanarayan. 2020. Visualint: Sketchy in situ annotations of chart construction errors. *Computer Graphics Forum*, 39(3):219–228.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *10th International Conference on Learning Representations, ICLR 2022*.

J. D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*.

Xingyu Lan and Yu Liu. 2025. "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.

Claire Lauer and Shaun O'Brien. 2020. The deceptive potential of common design tactics used in data visualizations. In *Proceedings of the 38th ACM International Conference on Design of Communication*, SIGDOC '20, New York, NY, USA. Association for Computing Machinery.

Maxim Lisnic, Cole Polychronis, Alexander Lex, and Marina Kogan. 2023. Misleading beyond visual tricks: How people actually lie with charts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Fangyu Liu, Julian Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. 2023. DePlot: One-shot visual language reasoning by plot-to-table translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10381–10399, Toronto, Canada. Association for Computational Linguistics.

Leo Yu-Ho Lo, Ayush Gupta, Kento Shigyo, Aoyu Wu, Enrico Bertini, and Huamin Qu. 2022. Misinformed by visualization: What do we learn from misinformative visualizations? In *Computer Graphics Forum*, volume 41, pages 515–525. Wiley Online Library.

Leo Yu-Ho Lo and Huamin Qu. 2025. How good (or bad) are llms at detecting misleading visualizations? *IEEE Transactions on Visualization and Computer Graphics*, 31(1):1116–1125.

Konrad J. Maciborski, Karolina Wysocka, Karolina Żelazowska-Byczkowska, Styliani Kleanthous, and Adam Wierzbicki. 2025. Boosting data literacy: The role of ai in teaching detection of deceptive charts. In *Artificial Intelligence in Education*, pages 92–105, Cham. Springer Nature Switzerland.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland. Association for Computational Linguistics.

Andrew McNutt and Gordon Kindlmann. 2018. 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*, volume 1, page 9.

Andrew McNutt, Gordon Kindlmann, and Michael Correll. 2020. Surfacing visualization mirages. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–16, New York, NY, USA. Association for Computing Machinery.

Shaun O'Brien and Claire Lauer. 2018. Testing the susceptibility of users to deceptive data visualizations when paired with explanatory text. In *Proceedings of the 36th ACM International Conference on the Design of Communication*, SIGDOC '18, New York, NY, USA. Association for Computing Machinery.

OpenAI. 2023. Gpt-4 technical report. Technical report, OpenAI.

Anshul Vikram Pandey, Anjali Manivannan, Oded Nov, Margaret Satterthwaite, and Enrico Bertini. 2014. The persuasive power of data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2211–2220.

Anshul Vikram Pandey, Katharina Rall, Margaret L. Satterthwaite, Oded Nov, and Enrico Bertini. 2015. 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, page 1469–1478, New York, NY, USA. Association for Computing Machinery.

Saugat Pandey and Alvitta Ottley. 2025. Benchmarking visual language models on standardized visualization literacy tests. *Computer Graphics Forum*, 44(3):e70137.

Qwen-Team. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, abs/2502.13923.

Jihyun Rho, Martina A Rau, Shubham Kumar Bharti, Rosanne Luu, Jeremy McMahan, Andrew Wang, and Jerry Zhu. 2023. Various misleading visual features in misleading graphs: Do they truly deceive us? In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46.

JonRobert Tartaglione and Lee de Wit. 2025. How the manner in which data is visualized affects and corrects (mis)perceptions of political polarization. *British Journal of Social Psychology*, 64(1):e12787.

Jonathan Tonglet, Tinne Tuytelaars, Marie-Francine Moens, and Iryna Gurevych. 2025. Protecting multimodal large language models against misleading visualizations. *arXiv preprint arXiv:2502.20503*, abs/2502.20503.

Edward R Tufte and Peter R Graves-Morris. 1983. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT, Cheshire, CT, USA.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Brenda W. Yang, Camila Vargas Restrepo, Matthew L. Stanley, and Elizabeth J. Marsh. 2021. Truncating bar graphs persistently misleads viewers. *Journal of Applied Research in Memory and Cognition*, 10(2):298–311.

Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. 2024. TinyChart: Efficient chart understanding with program-of-thoughts learning and visual token merging. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1898, Miami, Florida, USA. Association for Computational Linguistics.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, and 32 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, abs/2504.10479.

A Explanation of Misviz examples

Figure 5 shows the same misleading visualizations as in Figure 1, now overlaid with visual explanations highlighting the specific misleaders. We describe each case below:

(1) Misrepresentation In terms of percentages, Bernie Sanders is closer to Joe Biden than to Elizabeth Warren. However, the fourth bar (Sanders) is visually closer in height to the first (Warren) than the second (Biden), exaggerating the gap between Biden and Sanders.

(2) 3D The 3D effects make it difficult to compare values across years visually. For instance, it's unclear which year had the highest number of singles sold between 1995 and 1997.

(3) Truncated axis The vertical axis begins at 36%, exaggerating the gap between pro- and anti-Brexit responses.

(4) Inappropriate use of pie chart The pie chart shows responses to three overlapping time categories: today, last year, and 1997. Because respondents may fall into more than one category, the percentages exceed 100%. The pie chart suggests the categories are distinct.

(5) Inconsistent binning size The two bars compare economic growth over different time spans. The first covers a longer period than the second, making the comparison misleading.

(6) Discretized continuous variable Germany and Denmark have similar values, but are shown in different colors. Meanwhile, Austria shares Germany's color despite being further apart in value. The discrete color scale exaggerates differences between countries.

(7) Inconsistent tick intervals The dates on the horizontal axis are spaced unevenly. This distorts

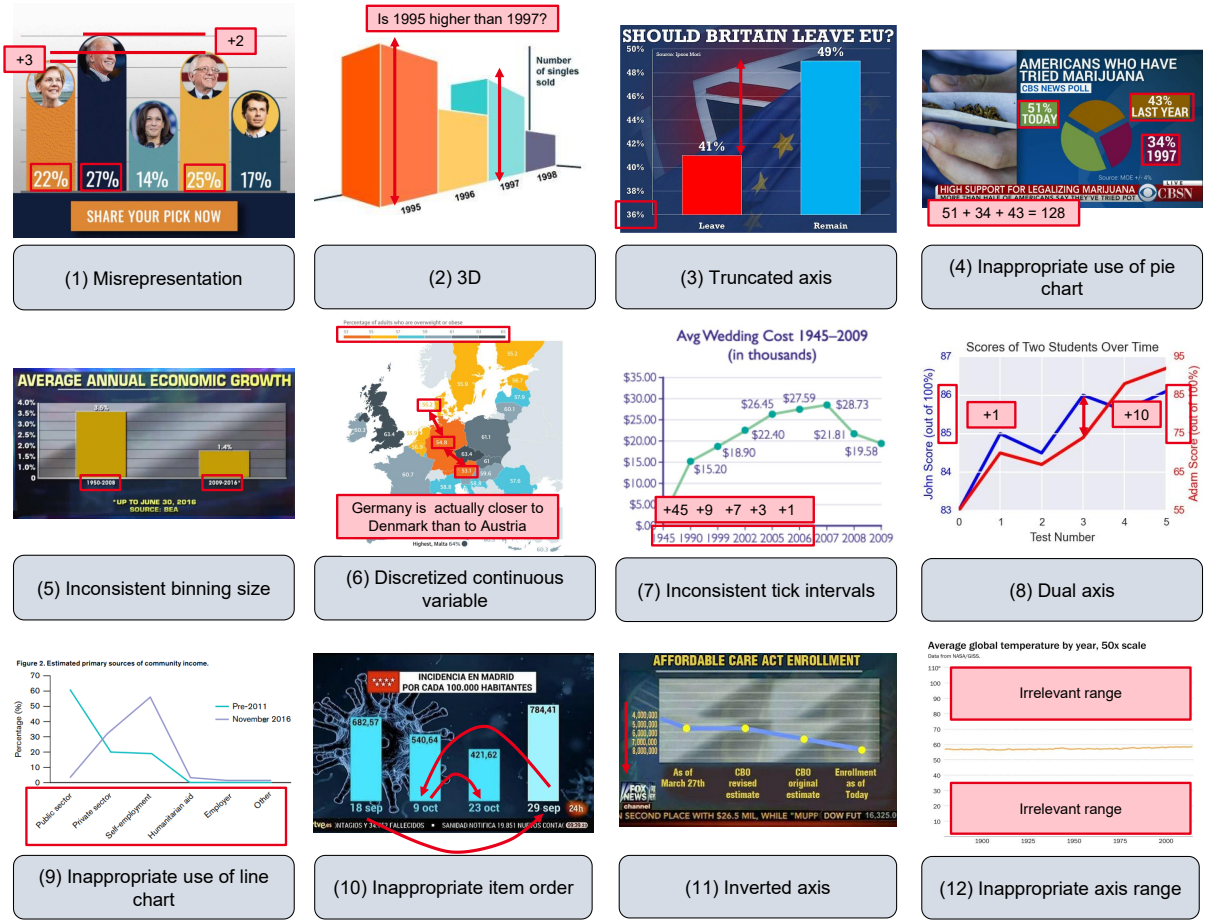


Figure 5: Examples of misleading visualizations from Figure 1, overlaid with visual explanations.

the slope of the trend line and gives a misleading impression of the progression of average wedding costs over time.

(8) Dual axis The scores for John and Adam are plotted on separate vertical axes. While the visual gap on test 3 looks small on the left axis, the actual difference is 11%, not 1%.

(9) Inappropriate use of line chart A line chart is used to connect values across a categorical variable. The line implies a trend that does not exist.

(10) Inappropriate item order The dates are out of chronological order, which can mislead to thinking COVID-19 cases initially dropped and then rose. In fact, the reverse is true.

(11) Inverted axis The vertical axis increases from top to bottom. At first glance, it seems that enrollment in the Affordable Care Act is falling over time, while it is actually rising.

(12) Inappropriate axis range The vertical axis is so broad that the trend in global average temperature appears nearly flat. A more appropriate narrow axis range would reveal a clearer upward trend.

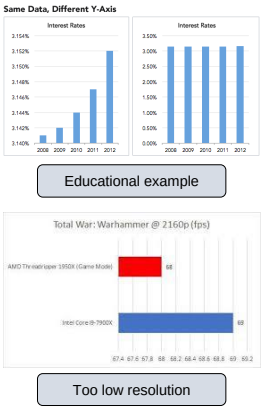


Figure 6: Examples of visualizations discarded from the corpus of Lo et al. (2022). Top: The misleading visualization is shown alongside a corrected version. Bottom: The visualization is too low-resolution.

B Example of discarded visualizations

Figure 6 provides two examples from the corpus of Lo et al. (2022) that were not included in Misviz.

Lan and Liu (2025)	Lo et al. (2022)
Data-visual disproportion	Misrepresentation
3D effect	3D
Truncated axis	Truncated axis
Misuse of part-to-whole relationship (pie chart)	Inappropriate use of pie chart
Uneven axis interval	Inconsistent tick intervals
Dual axis	Dual axis
Uneven data grouping	Inconsistent binning size
Categorical encoding for continuous data	Discretized continuous variable
Continuous encoding for categorical data (line chart)	Inappropriate use of line chart
Inverted axis	Inverted axis

Table 4: Mapping of misleaders from the taxonomy of Lan and Liu (2025) to the taxonomy of Lo et al. (2022).

Misleader category	Misviz	Misviz-synth
Misrepresentation	32.6	5.5
3D	14.0	2.2
Truncated axis	8.7	0.8
Inappropriate use of pie chart	6.2	1.2
Inconsistent tick intervals	5.1	15.8
Dual axis	3.0	3.0
Inconsistent binning size	2.8	2.1
Discretized continuous variable	1.9	4.0
Inappropriate use of line chart	1.8	2.2
Inappropriate item order	1.5	8.4
Inverted axis	1.2	11.9
Inappropriate axis range	1.1	4.0

Table 5: Percentage of instances per misleader.

C Mapping of misleader categories between taxonomies

Table 4 presents the manual mapping applied between the misleader categories from the taxonomy of Lan and Liu (2025) to those of Lo et al. (2022).

D Annotation interface and instructions

Figure 7 shows the main interface used by Prolific crowdworkers to label the visualizations from *r/dataisugly* and *r/dataisbeautiful*.

Figures 8, 9, and 10 show the labeling instructions given to the crowdworkers.

E Detailed dataset statistics

Tables 5 and 6 report the distributions of misleader categories and chart types in Misviz and Misviz-synth. The percentages do not sum to 100% in Misviz, because a visualization may contain more than one misleader category or chart type.

Of the 1,791 misleading visualizations in Misviz, 353 originate from the corpus of Lo et al. (2022), 964 were collected from the WTF Visualizations website (Lan and Liu, 2025), and the remaining 474 were sourced from *r/dataisugly* and annotated by crowdworkers.

Chart type	Misviz	Misviz-synth
Bar	37.0	21.6
Line	20.0	33.5
Pie	23.4	3.6
Map	4.9	16.9
Scatter plot	4.3	24.4
Other	16.5	0.0

Table 6: Percentage of instances per chart type.

Label	Precision
No misleader	80
Misrepresentation	40
3D	100
Truncated axis	20
Inappropriate use of pie chart	60
Inconsistent binning size	40
Discretized continuous variable	70
Inconsistent tick intervals	90
Dual axis	30
Inappropriate use of line chart	0
Inappropriate item order	20
Inverted axis	20
Inappropriate axis range	30

Table 7: Manual verification of the weakly labeled corpus annotated with Qwen2.5VL-72B (%).

F Examples of language and image type diversity in Misviz

Figure 11 shows four instances highlighting the diversity of Misviz. The two visualizations at the top have embedded text written in German and Estonian. The bottom visualizations are photographs of a visualization shown on a television screen and on a printed document.

G Extending Misviz with weakly labeled corpus

Lo et al. (2022) scraped a large collection of images to construct their taxonomy of visualization misleaders. The images obtained via web searches for keywords related to misleading visualizations. Only a small subset of this collection was manually annotated. In this section, we explore the potential of automatically scaling the dataset using this unlabeled collection.

We use a three-step process to create a weakly labeled corpus from the remaining unlabeled portion of the collection. First, we filter unreadable images with a heuristic: we extract texts from the

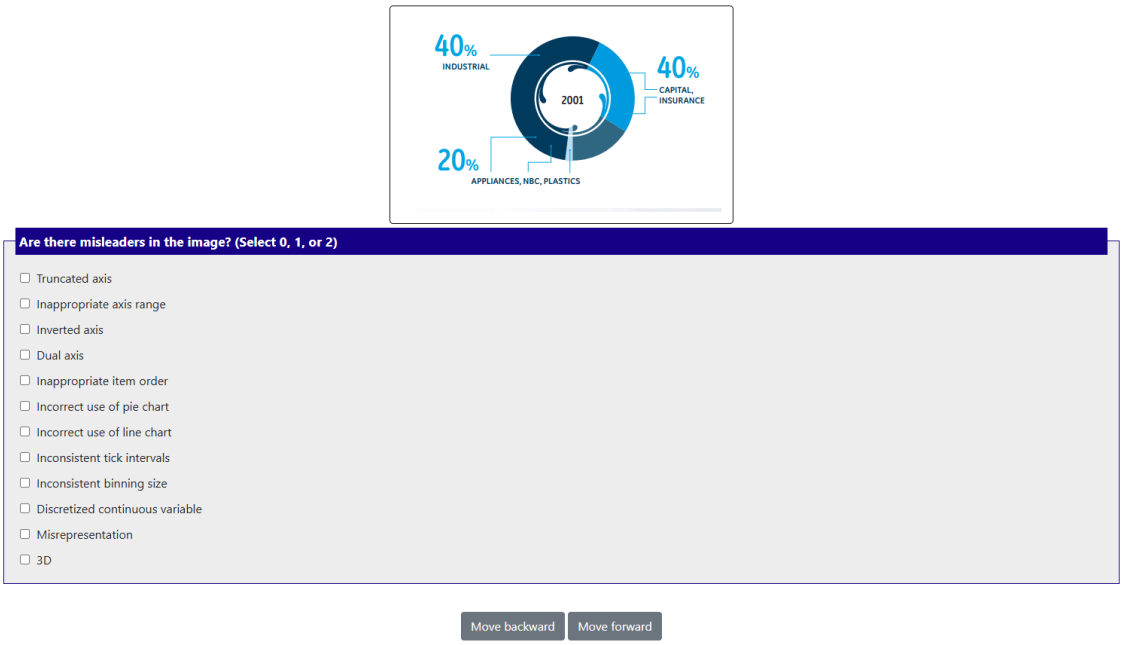


Figure 7: Main interface used by Prolific crowdworkers to assign misleader labels to a visualization.

Task 1: Chart types

In this study, you will be asked to classify a chart into the correct categories (pie chart, line chart, bar chart, map, scatter plot, or other). You are expected to spend, on average, 10 to 15 seconds on each chart.

Select one or more types of charts that appear in the image. You should always select at least one. You can choose multiple answers if the image contains more than one type.

Figure 8: Instructions for the first annotation task.

image using EasyOCR[6] and only keep images with an average OCR confidence over all detected text that is above 70. For the second and third steps, we use Qwen2.5VL-72B, the best performing open-weight model in terms of F1. In the second step, we ask the MLLM to determine whether the image depicts a visualization. The prompt is provided in Figure 13. Afterwards, we remove all images that do not depict a visualization. In the third step, we use the standard task prompt to detect which misleaders affect the visualization, if any, shown in Appendix K. We only keep images for which zero or one misleader is identified.

The resulting weakly labeled corpus contains 4,053 instances. We conducted manual verification of 10 instances per label, for a total of 130. As shown in Table 7, the results are uneven. For some misleaders, such as *3D* and *Inconsistent tick intervals*, Qwen2.5VL-72B achieves high precision on the manually verified sample, indicating that the corresponding subset of the corpus could be used to further train models to detect real-world cases of those misleaders. However, for other misleaders, such as *inappropriate use of line chart*, the precision is so low that the weak labels should not be considered reliable. Furthermore, we observed that several unreadable images remained in the corpus despite the OCR filtering step. In addition, some images are educational examples that explicitly state they show a misleading visualization, as explained in Appendix B. Removing these examples would require an additional step of human verification. Overall, the weakly labeled corpus cannot be considered a reliable dataset, except for a few misleaders. Future research should further explore how to scale the collection of real-world misleading visualizations.

H Misviz-synth column types

Table 8 lists the 10 column types used in the first step of the Misviz-synth synthetic visualization generation process. Each column is assigned exactly one primary type, which determines its core

[6] github.com/JaidedAI/EasyOCR

Task 2: Misleading visualizations

In this study, your task is to detect misleaders present in a chart. Misleaders are design flaws that decrease the ability of readers to interpret charts correctly. You are expected to spend, on average, 30 seconds to 1 minute on each chart.

Misleaders are chart design elements that can lead readers to interpret a wrong message that does not correspond to the underlying data. This study covers 12 types of misleaders. Each chart contains one to three misleaders (one being the most frequent case). Your task is to select the correct misleader(s) for each chart. We present the types of misleaders below.

Important remarks
- To perform this task, you need to pay attention to all aspects of the chart: axes and their labels, titles, legends, and proportions. Misleaders often reside in small details.
-It is essential to select the correct misleader(s) that apply to the image
Most charts contain only one misleader. In some cases, there will be two or three.

Figure 9: Instructions for the second annotation task.

Column type	Characteristic
Primary column types	
Temporal	Contains temporal values
Categorical	Contains categorical values
Numerical	Contains numerical values
Secondary column types	
Datetime	Datetime objects
Evenly spaced, unique temporal	Evenly spaced temporal values
Country	Country names
Unique object	Non-numerical values appear only once
Is part of whole	Numerical values sum to 1 or 100
Numerical percentage	Explicit percentage values (e.g., marked with %)
Potential percentage	Potential percentage values (e.g., value range is [0,1] or [0,100])

Table 8: Primary and secondary column types used during synthetic visualization generation

data role. Additionally, columns can be assigned one or more secondary types, which describe extra attributes that help determine appropriate chart types and potential misleaders.

I Examples of visualizations with axis metadata

Figure 12 provides two examples of visualizations from Misviz-synth with the corresponding axis metadata. The latter is represented as a table and stored as a JSON file with column names as keys.

J List of random design variations

All visualizations: color variation of the background, variations of the title template, variation of font type and size, and variation of chart size.

Bar and line charts: addition of minor ticks in addition to major ones, positioning of the vertical axis (left or right), variation in tick shapes, adding value labels on top of bars, variation of the tick step size, addition of chart borders, and addition of horizontal grid lines.

Bar charts only: sorting bars by values or by category name, hiding the vertical axis, placing labels on top of or within the bars, using horizontal or vertical value labels, variation of bar colors.

Line charts only: filling area below the line with a color, variation in line style and color, and addition of horizontal or vertical grid lines.

Pie charts only: placing data label next to the pie slices or placing them in a legend.

K Zero-shot prompts

Figure 14 shows the prompt to detect misleading visualizations. Figure 15 shows the variant used to generate bounding boxes, which is based on the prompt from Chen et al. (2025).

L Linter rules

We explain below the linter rules for detecting some misleaders in Misviz and Misviz-synth. These rules are implemented in Python and were designed on the validation set of Misviz-synth to maximize precision over recall.

Truncated axis A visualization has a truncated axis if one or more of its sorted vertical axes start

Task 3: Non-misleading visualizations

In this study, your task is to verify that a chart is not misleading, i.e., it does not have significant design flaws that decrease the ability of readers to interpret charts correctly. Your participation will help us build AI systems capable of classifying charts as misleading or not. You are expected to spend, on average, 30 seconds to 1 minute on each chart.

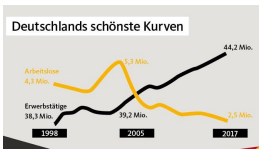
You will have to review charts that come from an online community. The majority of the charts are expected to be of high quality. However, some of them might be misleading. Misleaders are chart design elements that can lead readers to interpret a wrong message that does not correspond to the underlying data. There are 12 types of misleaders to consider, and they are explained below. If the chart is not misleading (i.e., in MOST CASES), you should leave the answer field blank. If the chart is misleading, select the misleaders that apply (up to three misleaders).

Important remark
- To perform this task, you need to pay attention to all aspects of the chart: axes and their labels, titles, legends, and proportions. Misleaders often reside in small details.
- You should not select a misleader for each chart! Most of them are not misleading. You should only detect the few that are misleading!

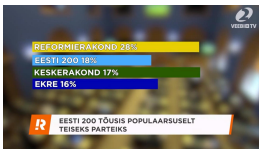
Figure 10: Instructions for the third annotation task.

	Misviz						Misviz-synth					
	Acc	Pre	Rec	F1	EM	PM	Acc	Pre	Rec	F1	EM	PM
Image	70.1 ± 0.9	73.7 ± 0.5	88.3 ± 1.8	58.7 ± 1.0	11.1 ± 0.7	14.9 ± 0.8	72.0 ± 0.5	71.9 ± 0.9	92.2 ± 1.3	64.7 ± 1.5	68.0 ± 1.2	68.0 ± 1.2
Image w. axis (DePlot)	72.8 ± 1.2	74.6 ± 0.6	92.0 ± 1.4	60.9 ± 1.4	12.3 ± 0.5	17.1 ± 0.5	72.5 ± 0.7	72.1 ± 0.6	92.6 ± 0.3	65.2 ± 1.2	69.5 ± 0.9	69.5 ± 0.9

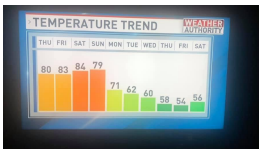
Table 9: Average performance with standard deviations (%) of the image-axis classifiers on the test sets of Misviz and Misviz-synth for instances with bar, line, or pie charts.



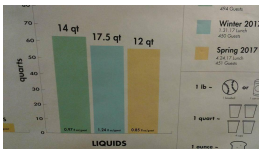
Visualization with text in German



Visualization with text in Estonian



Photograph of a visualization displayed on a television screen



Photograph of a printed visualization

Figure 11: Four visualizations from Misviz. Top: the embedded text is not written in English. Bottom: the visualization is not a screenshot but a photograph taken of an electronic device or printed material.

Figure 12: Two visualizations from Misviz-synth with their corresponding axis metadata.

with a numerical value strictly higher than 0.

Inverted axis A visualization has an inverted axis if one or more of its sorted axes is the reverse of the default axis order.

Dual axis A visualization has dual axes if it has two vertical axes with different axis tick labels.

Inappropriate item order A visualization has an inappropriate item order if it has a temporal axis with dates shuffled in random order. This covers only a specific case of inappropriate item order.

Prompt to detect visualizations

You are an expert in data visualization analysis. Your task is to identify whether the following image shows a data visualization or not. Provide only the final answer (visualization OR not a visualization), without additional explanation.

Figure 13: Prompt used to detect whether an image shows a visualization.

# Chart types	Image		Image w. axis (DePlot)		Linter	
	F1	PM	F1	PM	F1	PM
1 (1,956 instances)	58.2	15.2	60.7	17.4	35.3	7.9
2 (115 instances)	57.2	6.8	59.5	11.5	46.3	6.1
3 (6 instances)	77.5	0.0	38.9	0.0	33.0	0.0

Table 10: Performance of the image-axis classifiers and the linter on Misviz test (%) as a function of the number of chart types.

Inconsistent tick intervals A visualization has inconsistent tick intervals if the distance between the tick labels or relative positions on one or more of its numerical or temporal axes is not constant.

Inconsistent binning size A visualization suffers from inconsistent binning size if one axis shows categorical bins of unequal size.

M Results with standard deviations of image-axis classifiers

Table 9 reports the average results with standard deviations of the image-axis classifiers on the subset of Misviz with bar, pie, and line charts and on Misviz-synth.

N Additional implementation details

Training sets We fine-tune DePlot on the Misviz-synth train set. The train-small set is used to train the classification head.

DePlot fine-tuning We fine-tune DePlot for axis metadata extraction using two H110 GPUs. We retain the original DePlot architecture and loss function (Liu et al., 2023). The model is trained for 4 epochs using the Adam optimizer (Kingma and Ba, 2015), a learning rate of 5e-5, and a batch size of 4. To support memory-efficient training, we apply LoRA (Hu et al., 2022) on the query and value projection layers. To enhance generalization, we apply data augmentation through random rotations and perspective transformations.

# Chart types	Image		Image w. axis (DePlot)		Linter	
	F1	PM	F1	PM	F1	PM
1 (1,228 instances)	46.9	13.1	47.9	14.5	16.8	7.7
2 (203 instances)	46.4	24.3	48.1	32.7	17.8	8.9
3 (10 instances)	64.9	46.7	82.5	30.0	16.7	10.0

Table 11: Performance of the image-axis classifiers and the linter on Misviz test (%) as a function of the number of misleader types.

Image-axis classifiers The classification head is trained for up to 300 epochs with early stopping. We use a batch size of 256, a learning rate of 5e-5, and the Adam optimizer. To account for label imbalance, we apply a weighted loss based on the frequency of each misleader. The classification head has one hidden layer with 1.024 units. During training, we keep track of the best model weights. The best weights are updated at each epoch if the F1 score increases on the validation sets of both Misviz and Misviz-synth. While the classifier is not trained on Misviz, using the Misviz validation set ensures we do not overfit to the type of visualizations shown in Misviz-synth.

O Detailed Misviz results

Tables 10 and 11 report the performance of the image-axis classifiers and the linter on the Misviz test set as a function of the number of chart types and misleader types present in the visualization, respectively. These results provide further insights into the generalization gap between the Misviz-synth training data and the real-world visualizations of Misviz. Visualizations containing multiple chart types are substantially more challenging. This is expected, as multiple chart types increase visual complexity and introduce interactions that are not represented in Misviz-synth. In contrast to chart types, increasing the number of misleaders improves performance for both classifiers and for the linter. Visualizations with more misleaders increase the likelihood that at least one misleader is detected.

Table 12 reports the performance of MLLMs across three subsets of the Misviz test set, corresponding to its three sources of misleading visualizations. Performance varies significantly across subsets, but no single source is consistently easier or harder. This highlights the complementary value of each data source to the overall Misviz benchmark. Interestingly, the subset with the lowest EM is often consistent within the same model family,

Task prompt for misleader detection with zero-shot MLLMs

You are an expert in data visualization analysis. Your task is to identify misleaders present in the given visualization.

Please carefully examine the visualization and detect its misleaders. Provide all relevant misleaders, up to three, as a comma separated list. In most cases only one misleader is relevant. If you detect none of the above types of misleaders in the visualization, respond with "no misleader".

The available misleaders to select are, by alphabetical order:
- discretized continuous variable: a map displays a continuous variable transformed into a categorical variable by cutting it into discrete categories, thus exaggerating the difference between boundary cases.
- dual axis: there are two independent y-axis, one on the left and one on the right, with different scales.
- inappropriate axis range: the axis range is too broad or too narrow.
- inappropriate item order: instances of a variable along an axis are in an unconventional, non-linear or non-chronological order.
- inappropriate use of line chart: a line chart is used in inappropriate or unconventional ways, e.g., using a line chart with categorical variables, or encoding the time dimension on the y-axis.
- inappropriate use of pie chart: a pie chart does not display data in a part-to-whole relationship, e.g., its shares do not sum to 100%.
- inconsistent binning size: a variable, such as years or ages, is grouped in unevenly sized bins.
- inconsistent tick intervals: the ticks values in one axis are evenly spaced but their values are not, e.g., the tick value sequence is 10, 20, 40, 45.
- inverted axis: an axis is displayed in a direction opposite to conventions, e.g., the y-axis displays values increasing from top to bottom or the x-axis displays values increasing from right to left.
- misrepresentation: the value labels displayed do not match the size of their visual encodings, e.g., bars may be drawn disproportionate to the corresponding numerical value.
- truncated axis: an axis does not start from zero, resulting in a visual exaggeration of changes in the dependent variable with respect to the independent variable.
- 3d: the visualization includes three-dimensional effects.

Provide only the final answer, without additional explanation.

Figure 14: Task prompt for misleader detection with zero-shot MLLMs.

suggesting shared weaknesses in handling visualizations from the same data source.

P Prompt sensitivity analysis

We analyze the sensitivity of the two best-performing open-weight MLLMs to different input prompts. We consider three prompts: (1) the prompt only includes the name of the 12 misleader categories, (2) the prompt includes the definitions, (3) the default prompt, which includes the definitions and, in some cases, an example. The results for the Misviz val set, shown in Table 13, indicate a high level of sensitivity for EM and a moderate level for F1. The best prompt depends on the MLLM. Interestingly, InternVL3-38B achieves the highest EM when only the misleader names are provided, without any additional context.

Q Error examples

Figures 16 and 17 show examples of errors made by GPT-4.1 on the test sets of Misviz and Misviz-synth, respectively. One example is shown for each error category. Each example shows GPT-4.1's prediction in blue, the correct answer in green, and the error category in gray.

Task prompt for misleader detection with zero-shot MLLMs

You are given a chart (dimensions: {img_dim[0]} x {img_dim[1]}) with potential misleading regions:

Please analyze the image to detect misleaders and define bounding box coordinates for any misleading regions.
** Let's think it step by step! **
Here is the list of potential misleaders and their corresponding misleading regions:
- discretized continuous variable: a map displays a continuous variable is transformed into a categorical variable by cutting it into discrete categories, thus exaggerating the difference between boundary cases. The misleading region is the legend of the map.
- dual axis: there are two independent y-axis, one on the left and one on the right, with different scales. The misleading regions are the two vertical axes.
- inappropriate axis range: the axis range is too broad or too narrow. The misleading region is the vertical axis.
- inappropriate item order: instances of a variable along an axis are in an unconventional, non-linear or non-chronological order. The misleading region is (parts of) an axis.
- inappropriate use of line chart: a line chart is used in inappropriate or unconventional ways, e.g., using a line chart with categorical variables, or encoding the time dimension on the y-axis. The misleading region is one of the axis.
- inappropriate use of pie chart: a pie chart does not display data in a part-to-whole relationship, e.g., its shares do not sum to 100%. The misleading region is the labels on the pie slices.
- inconsistent binning size: a variable, such as years or ages, is grouped in unevenly sized bins. The misleading region is one of the axis or the legend for a map.
- inconsistent tick intervals: the tick values in one axis are not evenly spaced, e.g., the tick value sequence is 10, 20, 40, 45. The misleading region is (parts of) one of the axis.
- inverted axis: an axis is displayed in a direction opposite to conventions, e.g., the y-axis displays values increasing from top to bottom or the x-axis displays values increasing from right to left. The misleading region is one of the axis.
- misrepresentation: the value labels displayed do not match the size of their visual encodings, e.g., bars may be drawn disproportionate to the corresponding numerical value. The misleading region involves at least two objects (bar, pie slice) for which the size difference is not proportional to the value label difference.
- truncated axis: an axis does not start from zero, resulting in a visual exaggeration of changes in the dependent variable with respect to the independent variable. The misleading region is the starting tick of the axis. - 3d: the visualization includes three-dimensional effects. The misleading region is the 3D area of the chart.

Then output a JSON file containing coordinates for the potential misleaders and explanations.
*** Instructions:
- ** Please analyze the image (dimensions: {img_dim[0]} x {img_dim[1]}) to detect any misleading regions.**
- **Provide the misleading region coordinates with the name of the corresponding misleader**
- Your response format must strictly follow the example JSON format:
‘‘‘ ["coordinates": [[100, 200], [150, 200],[100, 300], [150, 300]],"misleader": "Truncated axis", "coordinates": [[250, 300], [300, 300],[250, 350], [300, 350]], "misleader": "Misrepresentation"]
‘‘‘

Figure 15: Task prompt variant for bounding box prediction.

	Corpus by Lo et al. (2022)			WTF Visualization (Lan and Liu, 2025)			*r/dataisugly*		
	Rec	EM	PM	Rec	EM	PM	Rec	EM	PM
Qwen2.5-VL-7B	47.1	7.6	17.9	24.7	**2.7**	3.4	31.8	7.2	7.2
Qwen2.5-VL-32B	96.0	9.2	12.6	96.9	**2.5**	3.3	93.5	2.7	2.7
Qwen2.5-VL-72B	93.7	31.5	55.5	82.8	**20.4**	21.5	83.2	32.5	32.5
InternVL3-8B	92.6	**18.1**	37.2	82.6	22.0	22.7	87.0	29.8	29.8
InternVL3-38B	71.8	23.7	49.2	53.9	34.6	35.5	51.7	**21.9**	21.9
InternVL3-78B	80.7	25.4	46.0	62.2	23.7	24.3	59.6	**21.6**	21.6
Gemini-2.5-Flash-Lite	86.8	33.1	60.8	52.8	**28.0**	28.4	74.0	**28.0**	29.7
GPT-4.1	96.2	**43.5**	73.1	94.2	58.8	60.2	91.4	57.9	57.9
GPT-o3	93.1	**50.4**	75.2	88.9	63.2	64.2	88.0	62.7	62.7

Table 12: Zero-shot MLLMs performance (%) on the test set of Misviz, separated by the data source of the misleading visualizations. The data source with the lowest EM for each MLLM is marked in **bold**.

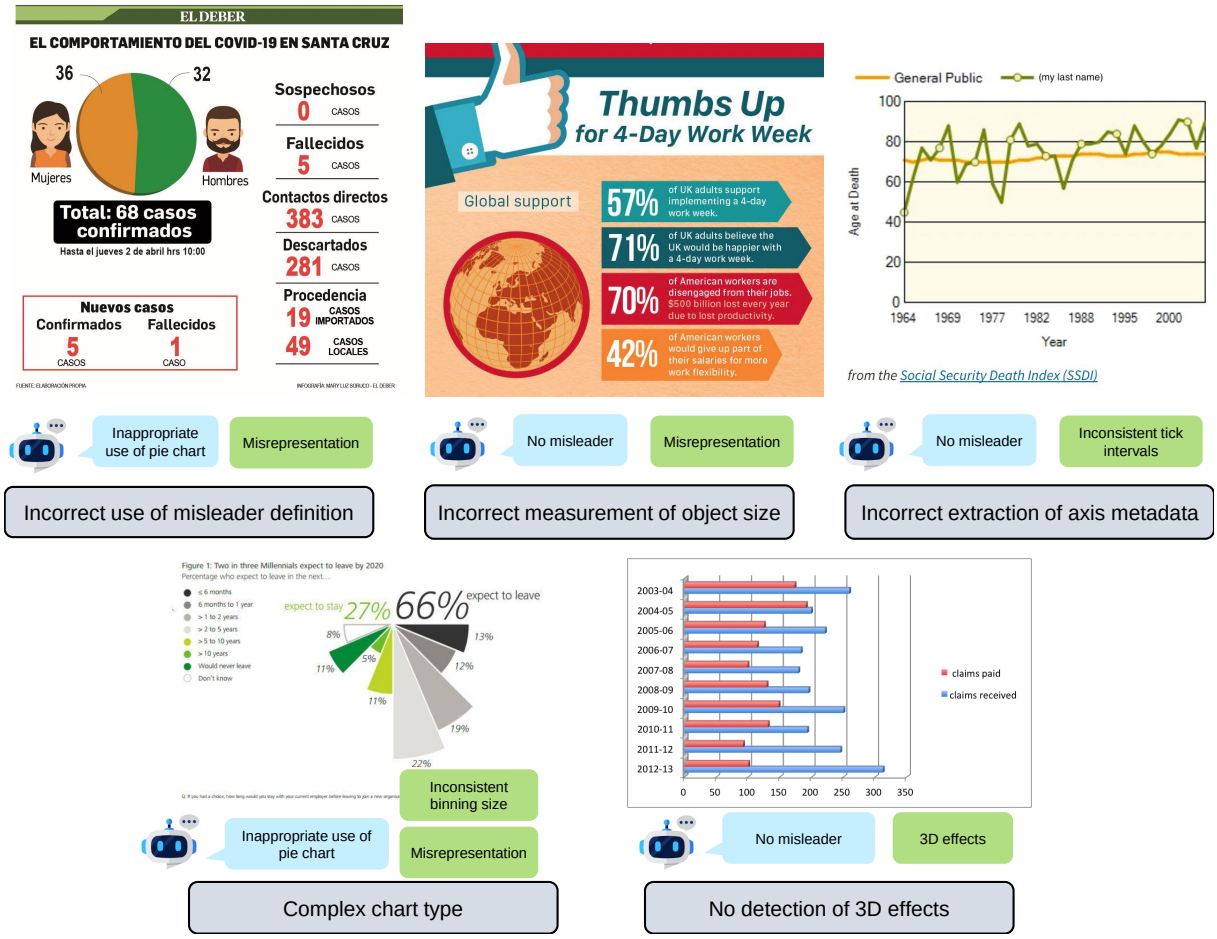
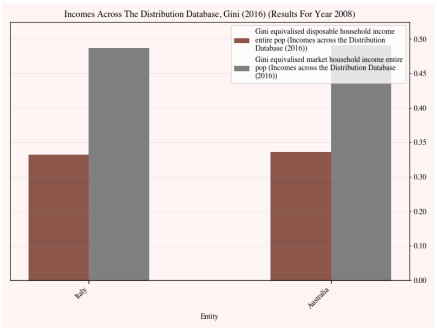
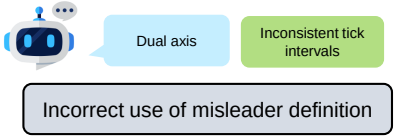
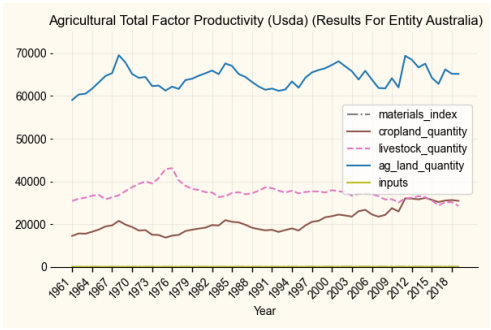


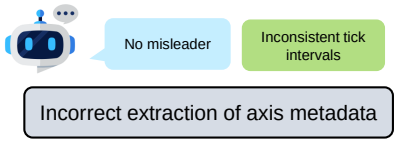
Figure 16: Error examples on the test set of Misviz. The predictions made by GPT-4.1 are shown in blue, while the ground truth is shown in green.



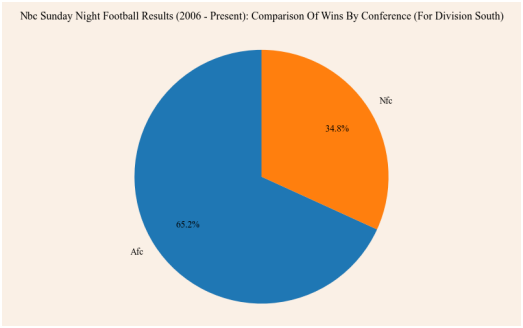


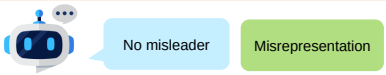
Incorrect use of misleader definition





Incorrect extraction of axis metadata





Incorrect measurement of object size

Figure 17: Error examples on the test set of Misviz-synth. The predictions made by GPT-4.1 are shown in blue, while the ground truth is shown in green.

Model	No definitions, no examples		Definitions, no examples		Definitions with one example	
	F1	EM	F1	EM	F1	EM
InternVL3-38B	54.3	29.2	58.4	25.5	58.5	26.6
Qwen2.5-VL-72B	61.2	15.7	62.8	23.2	62.0	22.1

Table 13: Prompt sensitivity analysis on the val set of Misviz (%).