Is This Chart Lying to Me?

Detecting Misleading Data Visualizations
Master thesis in Computer Science by Jan Zimny
Date of submission: May 5, 2025

1. Review: Prof. Dr. Iryna Gurevych
2. Review: Jonathan Tonglet
Darmstadt

TECHNISCHE
UNIVERSITÄT
DARMSTADT

UBIQUITOUS
KNOWLEDGE
PROCESSING

Computer Science
Department
Ubiquitous Knowledge
Processing Lab

Erklärung zur Abschlussarbeit gemäß §22 Abs. 7 APB TU Darmstadt

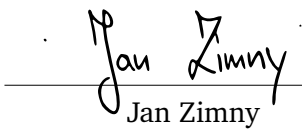
Hiermit erkläre ich, Jan Zimny, dass ich die vorliegende Masterarbeit gemäß §22 Abs. 7 APB TU Darmstadt selbstständig, ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt habe. Ich habe mit Ausnahme der zitierten Literatur und anderer in der Arbeit genannter Quellen keine fremden Hilfsmittel benutzt. Die von mir bei der Anfertigung dieser wissenschaftlichen Arbeit wörtlich oder inhaltlich benutzte Literatur und alle anderen Quellen habe ich im Text deutlich gekennzeichnet und gesondert aufgeführt. Dies gilt auch für Quellen oder Hilfsmittel aus dem Internet.

Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, den 5. Mai 2025



Jan Zimny

Acknowledgments

Parts of the text were linguistically revised with the help of Grammarly[1] and ChatGPT (OpenAI)[2] using prompts that focused on improving the flow and structure of the written text.

[1]Grammarly, `https://app.grammarly.com`, last visited on 5th of May, 2025
[2]ChatGPT-4o, `https://chatgpt.com`, last visited on 5th of May, 2025

Zusammenfassung

Datenvisualisierungen sind ein weit verbreitetes Medium zur Vermittlung komplexer Informationen, können jedoch durch schlechte Gestaltungsentscheidungen oder gezielte Manipulation auch irreführend sein. Diese Arbeit widmet sich der Herausforderung, irreführende Elemente in Datenvisualisierungen automatisch zu erkennen, indem ein auf synthetischen Daten trainierter, visuell basierter Klassifikationsansatz vorgeschlagen wird.

Zur Ermöglichung systematischer Experimente wird ein neuartiger, groß angelegter synthetischer Datensatz, *Misviz Synthetic*, eingeführt, der 14 Arten von irreführenden Elementen über Balken-, Linien- und Kreisdiagramme hinweg abdeckt.

Eine umfassende Ablationsstudie untersucht den Beitrag verschiedener Diagrammkomponenten, nämlich des Bildes, der Achsenmetadaten und der zugrundeliegenden Datentabelle, zur Klassifikationsleistung bei der Erkennung irreführender Visualisierungen. Die Ergebnisse zeigen, dass Achsenmetadaten komplementäre Informationen liefern, die von reinen Vision-Encodern häufig nicht erfasst werden, was insbesondere die Erkennung achsenbezogener irreführender Elemente verbessert.

Aufbauend auf diesen Erkenntnissen wird eine Klassifikationsarchitektur vorgeschlagen, die extrahierte Achsenmetadaten integriert und auf einem feinabgestimmten Modell des *Chart-to-Table*-Ansatzes *DePlot* basiert, genannt *Axis-DePlot*. Modelle, die auf *Misviz Synthetic* trainiert wurden, erreichen unter synthetischen Bedingungen eine starke Leistung. Das leistungsfähigste Modell kombiniert extrahierte Achsenmetadaten mit einer Bildrepräsentation und erreicht einen makro-gemittelten F_1-Score von 0,876. Bei der Evaluation auf dem realen Datensatz *Misviz* sinkt die Leistung jedoch deutlich, wobei das beste Modell einen durchschnittlichen Makro-F_1-Score von 0,188 erreicht. Diese Leistungslücke macht deutlich, wie wichtig es ist, die visuelle und strukturelle Vielfalt des synthetischen Datensatzes weiter zu erhöhen, um die Komplexität realer Datenvisualisierungen besser abzubilden.

Auch ohne vollständige Generalisierung stellt diese Arbeit einen wichtigen Schritt auf dem Weg zum langfristigen Ziel dar, Systeme zu entwickeln, die in der Lage sind, irreführende Diagrammelemente zu erkennen, zu erklären und gegebenenfalls zu korrigieren. Nach der Erkennung könnten solche Elemente genutzt werden, um Systeme zu unterstützen, die Erklärungen für irreführende Elemente liefern oder Korrekturmaßnahmen vorschlagen, wie etwa die Wiederherstellung einer abgeschnittenen y-Achse oder die Entfernung verzerrender Effekte. Letztlich trägt diese Forschung zur Entwicklung intelligenter Werkzeuge bei, die Nutzer dabei unterstützen, Datenvisualisierungen kritisch zu interpretieren und visueller Manipulation entgegenzuwirken.

Abstract

Data visualizations are a widely used medium for communicating complex information, but they can also be misleading due to poor design choices or intentional manipulation. This thesis addresses the challenge of automatically detecting misleading elements in data visualizations by proposing a vision-based classification approach trained on synthetic data.

To enable systematic experimentation, a novel large-scale synthetic dataset, *Misviz Synthetic*, is introduced, covering 14 misleader types across bar, line, and pie charts.

A comprehensive ablation study investigates the contribution of different chart components, namely the image, axis metadata, and the underlying data table, to misleading data visualization classification performance. The findings indicate that axis metadata provides complementary information that vision encoders alone often fail to capture, leading to improved detection of misleading elements, especially for axis-related misleaders.

Based on these findings, a classification architecture is proposed that incorporates extracted axis metadata using a model fine-tuned from the chart-to-table model *DePlot*, referred to as *Axis-DePlot*. Models trained on *Misviz Synthetic* achieve strong performance under synthetic conditions. The best-performing model utilizes extracted axis metadata combined with an image representation and achieves a macro-average F_1 score of 0.782. However, when evaluated on the real-world dataset *Misviz*, performance declines significantly, with the best model reaching a macro-average F_1 score of 0.188. This performance gap highlights the importance of further increasing the visual and structural diversity of the synthetic dataset to reflect the complexity of real-world data visualizations better.

Even without generalization, this work marks an important step toward building systems capable of identifying, explaining, and potentially correcting deceptive chart elements. Once detected, such elements could be used to inform systems that provide explanations for misleading design choices or propose corrective actions, such as restoring a truncated y-axis or removing distorting effects. Ultimately, this research advances the development of intelligent tools that assist users in critically interpreting data visualizations and resisting visual manipulation.

Acronyms

AUC Area Under Curve

CALVI Critical Thinking Assessment for Literacy in Visualizations

CC Creative Commons

CHARTOM CHARt Theory of Mind

CNN Convolutional Neural Network

CoT Chain-of-Thought

LLM Large Language Model

LM Language Model

LoRA Low-Rank Adapter

MLLM Multimodal Large Language Model

MLP Multi-Layer Perceptron

OCR Optical Character Recognition

OOD out-of-distribution

OWID Our World In Data

QA question answering

RMS Relative Mapping Similarity

RNN Recurrent Neural Network

RQ research question

SOTA state-of-the-art

SoViT Shape-Optimized Vision Transformer

Swin Shifted windows

ToMe Token Merging

ViT Vision Transformer

VLAT Visualization Literacy Assessment Test

WEF World Economic Forum

Contents

1. Introduction

1.1. Motivation

In our modern world, data visualizations have become an essential tool for conveying complex information quickly and clearly. For example, data visualizations, such as interactive maps, infographics, and statistical graphics, play a central role in shaping public understanding and guiding decision-making across domains such as politics, economics, and global affairs [1–4]. However, data visualizations can be misleading and intentionally manipulated to fit a particular political agenda and influence public opinion (see Figure 1.1). As a result, they may distort public understanding of currently unfolding events and contribute to conclusions that the underlying data does not support (see Figure 1.2) [5–7].

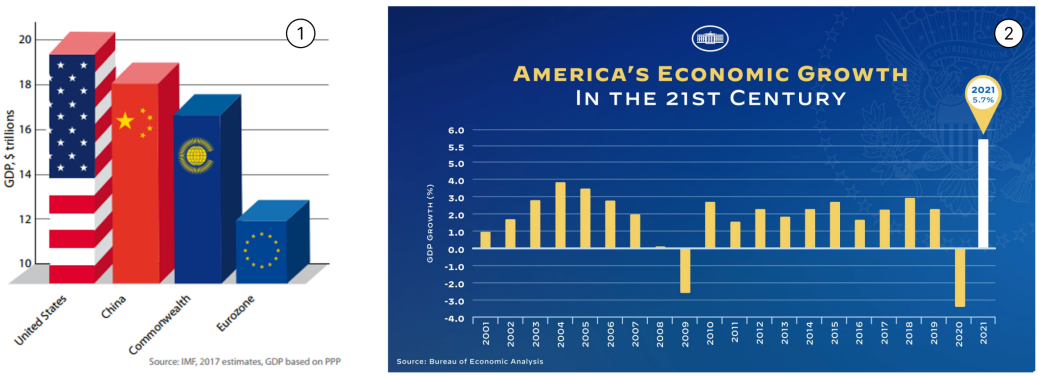


Figure 1.1.: **Examples of Misleading Data Visualizations.** ① The image shows a bar chart with a 3D effect applied and the y-axis is truncated, distorting the real proportions of the data [8]. ② The y-axis of the chart uses inconsistent y-axis scaling to make the emphasized value appear bigger than it is [9].

Given the potential for visualizations to mislead, whether through unintentional design flaws or deliberate manipulation, there is a growing need for tools to help evaluate their trustworthiness at scale. The research into multimodal misinformation detection has experienced rapid growth [11] due to the emergence of Multimodal Large Language Models (MLLMs), such as GPT-4 [12], Qwen [13], and InternVL [14], which possess advanced capabilities for interpreting both text and images and offer significant potential for automatically identifying misinformation and misleading data visualizations.

While research leveraging Large Language Models (LLMs) and MLLMs to detect text-based misinformation detection has picked up in recent years, the effectiveness of MLLMs on misleading data visualizations remains understudied, with prior efforts focusing on rule-based linting tools [15, 16]. Linting tools are based on the premise that misleading data visualizations typically stem from unintentional design decisions made by

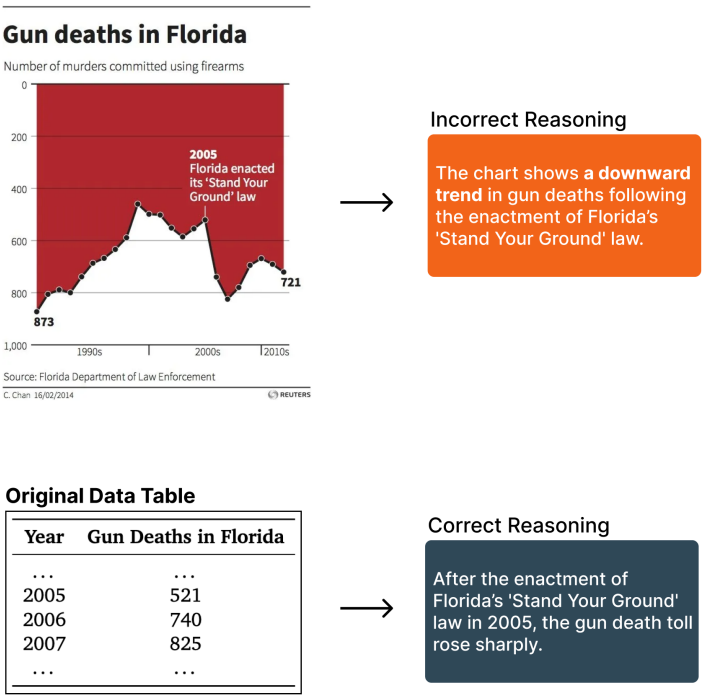


Figure 1.2.: **Illustration of a Reasoning Error Induced by a Misleading Data Visualization.** The y-axis is inverted, creating the false impression of a downward trend [10]. When comparing the chart to the actual data table, a viewer might arrive at two conflicting conclusions.

inexperienced users during the visual analytics process [16]. These approaches emphasize static analysis during the chart creation phase, aiming to identify potential sources of confusion or misinterpretation before the visualization is finalized and published [15, 16]. When access to the underlying data, the visualization code, and/or the chart specification is available, linting tools can automatically flag or correct misleading elements as part of the design workflow [15, 16]. However, in post-publication scenarios where only the final image is available, such tools are no longer practical, as they rely on access to the original data and generation context.

Consequently, recent studies have begun to investigate the capabilities of MLLMs in detecting misleading data visualizations, given only the image of the chart, by prompting the models to answer questions from datasets such as the Critical Thinking Assessment for Literacy in Visualizations (CALVI) [7], which probes human critical thinking skills in the context of deceptive visualizations, and CHARt Theory of Mind (CHARTOM) [17, 18], a dataset created to evaluate MLLMs capabilities regarding misleading data visualization detection and judgment of potential misleading factors. While CHARTOM primarily targets MLLMs, it is built upon principles of human critical thinking assessment [18]. However, the studies show that current state-of-the-art (SOTA) models struggle to identify misleading factors in charts in a question answering (QA) scenario [19, 20]. Other studies explicitly examine the capabilities of MLLMs in detecting misleading data visualizations by applying various prompting strategies [21, 22]. While MLLMs demonstrate promising performance in this area, their overall effectiveness still leaves room for improvement. Importantly, achieving higher accuracy often depends on the use of Chain-of-Thought (CoT) prompting [22] or the application of multiple prompts targeting individual misleading elements [21], suggesting that current models lack inherent robustness in reasoning about misleading data visualizations. Moreover, single-prompt approaches, where a model is asked to detect

all potential misleading elements at once, tend to scale poorly as the number of misleaders increases [22]. In addition, the limited size of existing datasets constrains current research to model evaluation [5–7, 17], hindering efforts toward robust training and generalization. Moreover, no dedicated benchmark exists for systematically comparing model performance in detecting misleading data visualizations.

To address these limitations, this thesis shifts the focus from prompt-based detection and reasoning to the direct detection of misleading elements in data visualizations. Rather than relying on carefully engineered questions or CoT prompting, we propose a vision-based classifier setup in which models directly identify misleading features in data visualizations. While this work focuses specifically on the detection task, this capability forms a critical foundation for combating misleading data visualizations. Accurate detection enables downstream systems to flag misleading data visualizations and serves as a necessary precursor to future developments, such as the automated explanation or correction of misleading data visualizations. To support this, this work introduces a large-scale synthetic dataset called *Misviz Synthetic*, which is specifically designed for training and evaluating models on misleading data visualization detection. This dataset enables systematic analysis of model capabilities and makes it feasible to train robust classifiers at scale. By advancing image-based detection with dedicated training data, this thesis takes an important step toward enabling tools that help viewers interpret visual data more critically, with the final goal of reducing the risk of people being misled by misleading data visualizations.

To address the research gap, this thesis makes the following key contributions:

- ***Misviz Synthetic:*** A novel large synthetic dataset based on real-world data covering 14 misleading visualization types across bar, line, and pie charts.
- **Ablation Study:** An extensive analysis across different vision encoders of which input features contribute the most to detecting misleading data visualizations.
- **Classification Model:** A novel model architecture that achieves SOTA performance in detecting misleading data visualizations on the *Misviz Synthetic* dataset.
- **Real-World Evaluation:** Assessment of the best-performing models from the ablation study on real-world misleading visualizations from the *Misviz* dataset.[1]

1.2. Research Goals

This thesis aims to advance the detection of misleading data visualizations by developing a synthetic dataset, *Misviz Synthetic*, which captures a diverse range of misleading data visualizations. *Misviz Synthetic* enables controlled experimentation to assess the impact of individual chart components on detecting misleading elements. While charts are often treated as visual artifacts, they are composed of multiple elements that can be exploited during analysis, such as the image itself, the axis labels and scales, and the underlying data table. This work analyzes which available features are most relevant for detecting misleading data visualizations. The synthetic dataset is used to identify which of these input modalities current chart vision encoders fail to capture effectively and whether incorporating them can enhance detection performance. In addition, models trained on synthetic data are evaluated for their generalization ability in real-world data visualizations. The following research questions (RQs) guide this work:

[1] Dataset in internal development by the UKP lab of the Technical University Darmstadt.

- **RQ1**: Which feature of a chart (the image itself, the axis metadata, or the underlying data table) contributes to the detection of misleading factors?
- **RQ2**: Can a trained model detect misleading data visualizations?
- **RQ3**: Can models trained on synthetic misleading charts generalize to real-world charts?

2. Related Work

2.1. Misinformation

The spread of misinformation is on the rise, with the World Economic Forum (WEF) listing it as the fourth most significant risk in its annual global risks report. According to the WEF, misinformation and disinformation have the potential to "[...] fuel instability and undermine trust in governance, complicating the urgent need for cooperation to address shared crises." [23] Misinformation is seen as false information, and disinformation is seen as "[...] deliberately false information." [24] Until recently, the study of misinformation has been focused on textual misinformation, with the study of misleading data visualizations being disregarded [25]. The growing ability of MLLMs to jointly reason over images and text [12–14] has given rise to new research directions within the broader study of misinformation. One such area is multimodal fake news detection and explanation, which focuses on identifying and interpreting deceptive news content that combines textual and visual information [26, 27]. Another emerging research direction is fake image detection and reasoning, which explores the identification of manipulated or counterfeit images and the explanation of their deceptive elements [28–30]. Compared to other forms of visual misinformation, research in misleading data visualizations is still in its early stages, with fewer studies addressing their detection and interpretation.

2.2. Misleading Data Visualizations

2.2.1. Definition of Misleading Data Visualizations

Misleading data visualizations are defined with varying terminology across scientific literature. Pandey et al. [31] define a "deceptive visualization" as "[...] a graphical depiction of information, designed with or without an intent to deceive, that may create a belief about the message and/or its components, which varies from the actual message." Fan et al. [32] employ the same term and define it similarly as "[...] visualizations that, whether intentionally or not, lead the reader to an understanding of the data which varies from the actual data." McNutt et al. [16] introduce the term "visualization mirage" to describe "[...] any visualization where the cursory reading of the visualization would appear to support a particular message arising from the data, but where a closer re-examination of the visualization, backing data, or analytical process would invalidate or cast significant doubt on this support." Lo et al. [5] use the term "misleading visualization" specifically for visualizations that are manipulated to appear supportive of claims not supported by the underlying data. Ge et al. [7] define the term "misleaders," which depicts "[...] decisions made in the construction of visualizations that can lead to conclusions not supported by the data."

Analyzing these definitions reveals key commonalities:

- a potential discrepancy between the impression created by the visualization and the actual data,

- the potential to be both intentionally and unintentionally misleading, and
- the result of viewers forming incorrect interpretations.

Based on the mentioned factors, this work utilizes the term *misleading data visualizations*, defined as visualizations that, whether created intentionally or unintentionally, lead viewers to an understanding or conclusion about the data that deviates from the underlying or real-world data, which results in the viewer potentially forming incorrect interpretations about the data. Furthermore, to differentiate misleading data visualizations from the factors that cause them, this work utilizes the term *misleader*. The term is defined as the factor that leads visualizations to become misleading data visualizations. Furthermore, the term *chart* is used as a synonym for *data visualizations*.

2.2.2. Taxonomy of Misleaders

Although misleading data visualizations have been studied for decades, this discussion centers on work that comprehensively categorizes and explains a broad range of misleader types. One notable example is the work by Lo et al. [5], who present a foundational study on misleading visualizations. The work analyzes over one thousand real-world examples of visualizations reported as deceptive or misleading and uses open coding to identify 74 distinct types of misleaders in data visualizations. Based on the found misleaders, Lo et al. developed a detailed taxonomy of misleading elements in visualizations. This taxonomy offers a structured way to understand and categorize misleading factors in data visualizations. McNutt et al. [16] introduce a comprehensive overview of errors in the design process of a chart, which can result in misleading data visualizations, categorized by the design step in which the error might arise. Lan et al. [33] propose a new taxonomy built on top of previous misleader taxonomy studies [5, 6, 16], identifying new misleading factors in charts and analyzing future research areas in the field.

2.2.3. Risk Posed by Misleading Data Visualizations

Given the diverse range of misleaders, quantifying their risk remains challenging. In the following sections, existing literature on two key aspects is reviewed: first, the frequency with which misleading visualizations occur in real-world contexts, and second, the documented impacts these visualizations have on viewer perception.

Frequency of Misleading Data Visualizations
There is limited empirical research that systematically quantifies the frequency of misleading data visualizations in real-world settings. One exception is the study by Lisnic et al. [6], which analyzed visualizations collected from the official Twitter COVID-19 streaming endpoint (now referred to as X). The authors report that 12% of the sampled visualizations exhibited characteristics classified as misleading. The most commonly identified misleader types were the use of dual axes (5.4%), area or *3D* encodings (5.0%), and *truncated axes* (1.2%).

Impact of Misleading Data Visualizations
Data visualizations have been shown to significantly affect changes in opinion and excel at amplifying related messages [34]. Undetected misleading data visualizations have the potential to amplify or alter the viewer's opinion. Due to the wide variety of misleaders, studies on the impact of misleading data visualizations typically focus on selected misleading factors.

Pandey et al. [31] analyzed how misleading data visualizations affect data interpretation. Their study examined both "exaggerated message" misleaders, which exaggerate the message the underlying data communicates

(*truncated y-axis* in bar charts, *area encoding* in bubble charts, and *distorted aspect ratio* in line charts), and a "message reversal" misleader, which inverts the message of the underlying data (*inverted y-axis* in line charts). Among 250 participants rating perceived differences between values on a 5-point 5 Likert scale, those shown misleading visualizations perceived differences as 58.5 to 129.5% larger compared to those shown non-misleading visualizations. In the message reversal test, 97.5% of participants presented with the misleading chart incorrectly interpreted whether values had improved or declined.

Rho et al. [18] conducted an empirical study to assess the impact of 14 types of misleading graphs on viewers' ability to interpret data. Based on a sample of 78 undergraduate students, the study found that misleading charts significantly reduced participants' accuracy in interpreting data values compared to non-misleading versions. While some chart types, such as *inverted axes* or *manipulated time intervals*, strongly impaired comprehension, others, like pictorial bars or compressed y-axes, had minimal effect. The study demonstrates that not all misleaders are equally impactful and emphasizes prioritizing the most harmful types in educational interventions.

Prior work by Smith et al. [35] investigated the impact of misleading data visualizations when paired with accurate textual descriptions. Through an online survey with 256 participants evenly divided between control and test groups, they found that misleading data visualizations successfully misled viewers, even if the paired textual description was factually accurate.

In the line of study of human literacy of data visualizations, Ge et al. [7] published CALVI, an assessment test for humans consisting of 45 questions, which combine multiple-choice and true-or-false formats, designed to measure "[...] people's ability to read, interpret, and reason about erroneous or potentially misleading visualizations". CALVI's structure incorporates 15 visualizations from the Visualization Literacy Assessment Test (VLAT), which features non-misleading visualizations [36], alongside 15 visualizations intentionally containing misleaders. The test covers nine data visualization types and 11 misleader types. To validate the effectiveness of CALVI, the researchers conducted a trial study with 497 participants, analyzing item easiness, item discrimination metrics, and correct answer rates. The assessment revealed patterns in how likely people are to identify specific misleaders, providing a comprehensive overview of the relative difficulty of detecting various deceptive techniques in data visualizations [7].

2.3. Automated Chart Understanding

Automated Chart Understanding is an interdisciplinary area that applies computer vision and natural language techniques to interpret data visualizations. The field focuses on tasks such as question answering [37–41], chart-to-table conversion [42, 43], and summarization [44–47] (A comprehensive overview is provided by Huang et al. [48]).

2.3.1. Background on Transformers and Vision Encoders

The transformer, introduced by Vaswani et al. [49] in the paper "Attention Is All You Need", is a deep neural network architecture. The architecture relies on multi-head self-attention to model long-range dependencies by weighing relationships between all input elements. Compared to prior model architectures such as Convolutional Neural Networks (CNNs) [50] and Recurrent Neural Networks (RNNs) [51, 52], transformers offer significantly increased parallelization and training efficiency [49]. The architecture has since become the predominant architecture in SOTA LLMs and MLLMs [12–14] and has been adapted to computer vision.

Dosovitskiy et al. [53] introduced Vision Transformers (ViTs), treating images as sequences of patch embeddings processed by a transformer encoder. Given sufficient training data, this approach matches or surpasses CNNs on image classification tasks.

Beyond the ViT architecture, several alternative architectures have been proposed to improve the efficiency and scalability of vision-based transformers. One such approach is the Shape-Optimized Vision Transformer (SoViT) [54] architecture, a lightweight vision transformer that processes images as smaller, non-overlapping slices to reduce memory usage and computation while preserving spatial structure. Also of note is the Shifted windows (Swin) [55] transformer architecture, which introduces a hierarchical structure that restricts attention computation to non-overlapping windows and shifts these windows between layers. As a result, Swin achieves linear computational complexity with respect to image size and demonstrated SOTA performance across image recognition and dense prediction tasks at the time of publication [55].

2.3.2. Domain-Specific Models

Transformer-based encoders have proven effective in the automated chart understanding domain for advanced understanding tasks. Chart images are inherently multimodal (combining graphical elements with textual labels). Early approaches in the area of chart understanding utilized preexisting Optical Character Recognition (OCR) tools to extract the underlying data, graphical elements and textual labels of the chart, which were then used as input to a Language Model (LM) to perform domain-specific tasks [39, 42, 56–59]. Recent studies have utilized end-to-end trained vision language encoder-decoder models that learn structured representations of chart images, outperforming previous approaches [60–66]. All models deploy one [60–63, 65, 67] or multiple visual encoders [64, 67] (such as a CNN [50], ViT [53], SoViT [54] or Swin [55] transformer) to transform image inputs into dense feature representations, which are then used alone, or in combination with encoded textual inputs, and passed into a textual decoder to generate a desired answer. Because the encoded images represent the information required to complete the tasks they were fine-tuned on, they can be utilized for downstream tasks such as image classification.

2.4. Misleading Data Visualization Detection and Evaluation

2.4.1. Available Misleading Data Visualization Datasets

Training and evaluating models requires data. Currently, available datasets are mostly small and primarily used to evaluate misleading data visualization detection capabilities. Lo et al. [5] utilized open coding to discover various misleader types in data visualizations, annotating 1.142 images of data visualizations with 74 different misleading factors. The authors searched search engines and social media websites for terms related to misleading data visualizations to find suitable images. Additionally, the authors scraped the Reddit page */r/dataisugly*,[1] in total accumulating 129.125 data visualizations which contain potentially misleading factors. Around 6.500 images were analyzed to derive the misleader taxonomy.

The analysis by Lisnic et al. [6] of tweets from the official Twitter COVID-19 streaming endpoint (now referred to as X) yielded a dataset containing 1.276 annotated misleading data visualizations across seven misleader types.

[1] `https://www.reddit.com/r/dataisugly/`

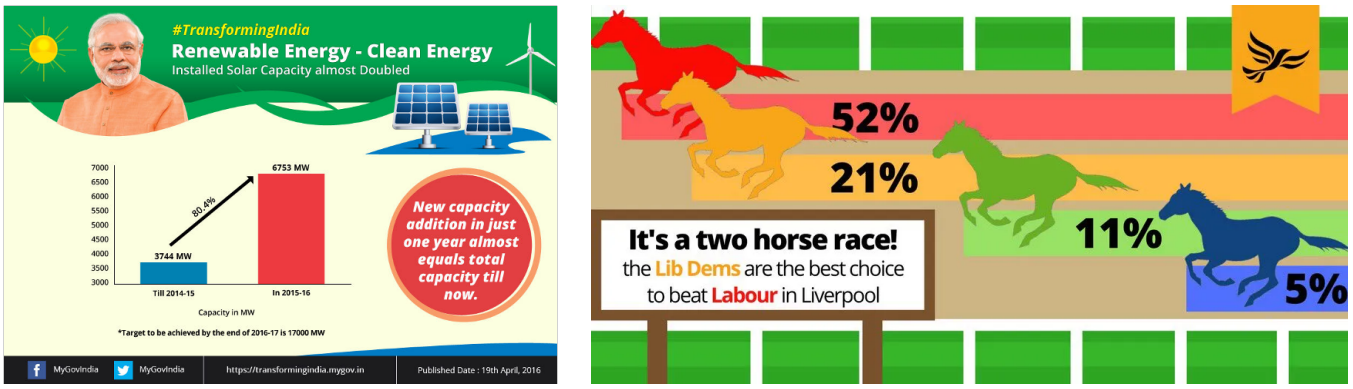


Figure 2.1.: **Two Examples from the *Misviz* Dataset.** The left image shows a line chart containing the misleader *truncated y-axis*, which makes the value on the right appear disproportionately larger than the other values. The right image depicts a horizontal bar chart with a *misrepresentation* misleader, as the bar sizes do not correspond to their respective value labels.

CHARTOM [17] is a QA benchmark to evaluate MLLMs perception of factual truth in data visualizations and the ability to recognize if a given data visualization will mislead a human. Built upon a human critical thinking assessment [18], the dataset consists of 112 data visualization images, which come in 56 pairs, with one chart being misleading and the other not. For each pair, a factual question and a reasoning question are posed. The factual question requires a one-word answer. The reasoning question includes multiple choice, free text entry, or sorting. The dataset covers pie-, bar-, line charts, scatter plots, and maps. The dataset covers 11 misleader types.

Ge et al. [7] developed a human critical thinking test that encompasses a question corpus of 60 data visualizations, each having a related true-or-false or multiple-choice question. In total, 45 of the contained charts are misleading. The other 15 contain no misleaders.

Although not associated with a peer-reviewed publication, the *MISCHA-QA* dataset[2] introduces a set of five misleader types, called *Non-Zero Baseline*, *Inconsistent Time Intervals*, *Over-Segmentation*, *Non-Sum to 100*, and *Non-Sum to 100%*, spanning three common chart types: bar, line, and pie charts. While *MISCHA-QA* is the largest publicly available dataset of its kind, comprising 8.205 images and including a dedicated training split, the process by which the data was generated remains undocumented, limiting the ability to assess its quality, potential biases, and suitability for evaluation tasks.

This work utilizes the *Misviz* dataset,[3] which encompasses 2.627 images of data visualizations. The dataset includes 12 misleader types, six chart types, and a diverse *Other* category. Overall, the dataset consists of annotated images of misleading data visualizations collected from prior studies [5, 33], additional self-annotated examples from the Reddit community */r/dataisugly*,[4] and non-misleading visualizations from */r/dataisbeautiful*.[5] Example data are shown in Figure 2.1.

[2] `https://huggingface.co/datasets/chart-misinformation-detection/MISCHA-QA`
[3] Dataset in internal development by the Ubiquitous Knowledge Processing lab of the Technical University Darmstadt.
[4] `https://www.reddit.com/r/dataisugly/`
[5] `https://www.reddit.com/r/dataisbeautiful/`

2.4.2. Dedicated Automatic Detection Approaches

The research on the automatic detection of misleading data visualizations is limited. Only one known work has developed a dedicated detection tool for misleading data visualizations. Fan et al. [32] propose a tool to annotate misleading factors in line charts, covering the misleaders *truncated y-axis*, *inverted y-axis*, and *distorted aspect ratio*. The approach utilizes OCR tools to extract line chart information and subsequently annotate and correct the misleading factors in the chart image. Fan et al. [32] used an experiment similar to Pandey et al. [31] to evaluate the tool. Participants were presented with line charts containing "exaggerated message" and "message reversal" misleaders (*truncated y-axis* and *inverted y-axis*). Instead of showing the non-misleading version of the chart, the authors employ the proposed annotation tool. Participants with access to the annotation tool were less likely to be impacted by the "message reversal" (*truncated y-axis*) and "message inversion" (*inverted y-axis*) techniques [32].

2.4.3. Evaluation of Detection Capabilities of Multimodal Large Language Models

In addition to dedicated detection approaches, recent studies have evaluated the detection capabilities of general-purpose MLLMs. Studies show that MLLMs have limited ability to detect misleading data visualizations.

Alexander et al. [21] utilize a subset of the COVID-19 tweet-visualization dataset from Lisnic et al. [6] to study the ability of the proprietary MLLMs GPT-4o mini [68], GPT-4o [69], and GPT-4V [70] to detect reasoning misleaders and visual misleading data visualizations. The models were tested under four prompting strategies, ranging from naive zero-shot to prompts with definitions and examples. Results show that the models can moderately detect misleading visuals without training, and performance improves with guided prompts. Alexander et al. [21] evaluated the models in a binary setting, asking whether a given misleader is present in the image, once per misleader type. The GPT-4o [69] model performs best with an AUC-score of 0.821. According to Alexander et al., "[...] a single prompt engineering technique does not yield the best results for all misleader types". The best prompting strategies depend on the misleader type: Definitions with examples are more effective for reasoning-based misleaders, whereas simple definitions suffice for visual misleaders [21].

In comparison, Lo et al. [22] use single prompt engineering techniques. Building on a dataset of misleading charts compiled in prior research [5], the authors evaluated the capabilities of four MLLMs using a set of nine prompts varying in complexity. Across three stages of experimentation, they expanded the scope of data visualization misleader types from 5 to 21 categories. They employed different prompting techniques (CoT, dynamic/split CoT, direct JSON/textual output). The authors demonstrate that as the number of misleaders increases, prompt size grows and requires careful design to maintain prediction efficiency. The results indicate that MLLMs exhibit potential for chart comprehension and misleading data visualization detection skills [22].

Tonglet et al. [19] and Pandey et al. [20] evaluate the performance of multiple MLLMs QA skills on human data literacy and critical thinking tests. The two works utilized the VLAT [36] and CALVI [7] tests, with Tonglet et al. additionally including the CHARTOM [17] test. Both studies conclude that the evaluated MLLMs struggle to detect misleading data visualizations in a QA setting. This is particularly relevant as these tasks implicitly assess a model's ability to recognize deceptive or misleading elements in visualizations, even without being explicitly prompted to do so.

3. Creation of the Synthetic Misleading Data Visualization Dataset

A well-curated dataset is essential to train models capable of detecting misleading data visualizations. This chapter introduces the synthetic dataset *Misviz Synthetic*, which is based on real-world data and designed to systematically capture a wide range of misleader types. It outlines the dataset creation process, including the derivation of misleader types, the chart generation pipeline, resulting dataset characteristics, and the potential limitations.

3.1. Advantages and Drawbacks of Using Synthetic Datasets

While real-world datasets offer the advantage of reflecting the complexity and diversity of naturally occurring visualizations, they are often time-consuming and expensive to annotate [48]. In contrast, synthetic datasets allow complete control over parameters such as created chart types and the specific misleaders introduced. This enables the efficient generation of large volumes of labeled data. However, synthetic datasets also come with notable drawbacks. They may fail to capture the complexity, imperfections, and noise inherent to real-world visualizations. Special care must be taken to introduce meaningful variation and realism in the synthetic charts to address this. Ensuring sufficient diversity is essential to enable generalization beyond the training domain [48].

3.2. General Requirements

To enable robust training and evaluation of detection models, the synthetic dataset for misleading visualization detection should fulfill the following general requirements in this work:

- **Include both misleading and non-misleading visualizations:** This is essential for enabling supervised learning and meaningful performance evaluation.
- **Be sufficiently large:** The dataset should support standard machine learning workflows, including training, validation, and testing phases.
- **Cover a broad range of misleader types:** While the dataset does not aim to replicate real-world misleader distributions, it should ensure balanced and comprehensive representation of the selected misleader types.

- **Be grounded in real-world data:** To enhance the plausibility and authenticity of synthetic visualizations, the dataset should be generated using data from real-world, open-domain sources that explicitly permit modification, sharing, and publication. The use of real-world data is assumed to offer the following benefits:
  - It increases the alignment of synthetic visualizations with the structural and semantic characteristics commonly found in real-world visualizations.
  - It supports better model generalization by exposing models to realistic data distributions, variability, and noise.
  - It reduces the risk of generating implausible visual patterns that may result from overly simplistic or artificially constructed data.
- **Use a configurable charting framework:** All visualizations should be created using a plotting library that allows fine-grained control over chart elements.

3.3. Choice of Misleader and Chart Types

The synthetic dataset is designed to cover a representative range of chart types and misleader types. The selection is informed by existing literature on chart taxonomy and misleading data visualizations to ensure relevance and coverage. The derivation of chart and misleader types closely follows the methodology proposed by Ge et al. [7].

The misleaders covered in the synthetic dataset are derived from the works of Lo et al. [5] and McNutt et al. [16], both of which offer comprehensive and in-depth taxonomies of misleaders in data visualizations.

Misleader and Chart Type Derivation Process
The derivation process began by extracting the misleader subcategories proposed by McNutt et al. [16]. These were then merged with misleader types identified by Lo et al. [5], followed by a filtering step to retain only those that are visually detectable and not solely based on cognitive biases. Cognitive bias is a "[...] cognitive phenomenon which involves a deviation from reality that is predictable and relatively consistent across people." [71]. This means that an inherent bias in the viewer, which deviates from reality, distorts the perception of the data visualization and leads to incorrect reasoning. As an example, the categories *Biases in Interpretation* and *Base Rate Bias* [16] were removed. Similar misleader types were grouped and duplicates were removed (Ⓐ in Figure 3.1). For instance, *dual axis* and *truncated y-axis* are included in the taxonomy of both works [5, 16]. An example of grouped misleaders is the summarization of all misleaders under the category *incomplete chart* by Lo et al. [5], such as *missing title* and *missing legend*, into a single category named *incomplete chart*.

Next, misleader types that appear infrequently in real-world examples were excluded. To achieve this, frequency data from Lo et al. [5] was used, and all misleader types occurring three times or fewer were discarded. Where misleaders were grouped, their frequencies were summed, and irrelevant or duplicate entries within the annotated data were removed (Ⓑ in Figure 3.1). For example, *distractive value labels* and *inappropriate aspect ratio* were removed, as they occurred three times or fewer, making them infrequent. Lastly, misleaders were excluded if their misleading effects are easily perceivable by viewers, alerting them that the visualization may not accurately reflect the underlying data. To illustrate, the misleader *cluttering* was removed. While this misleader may cause reasoning errors when viewers attempt to read exact values, it

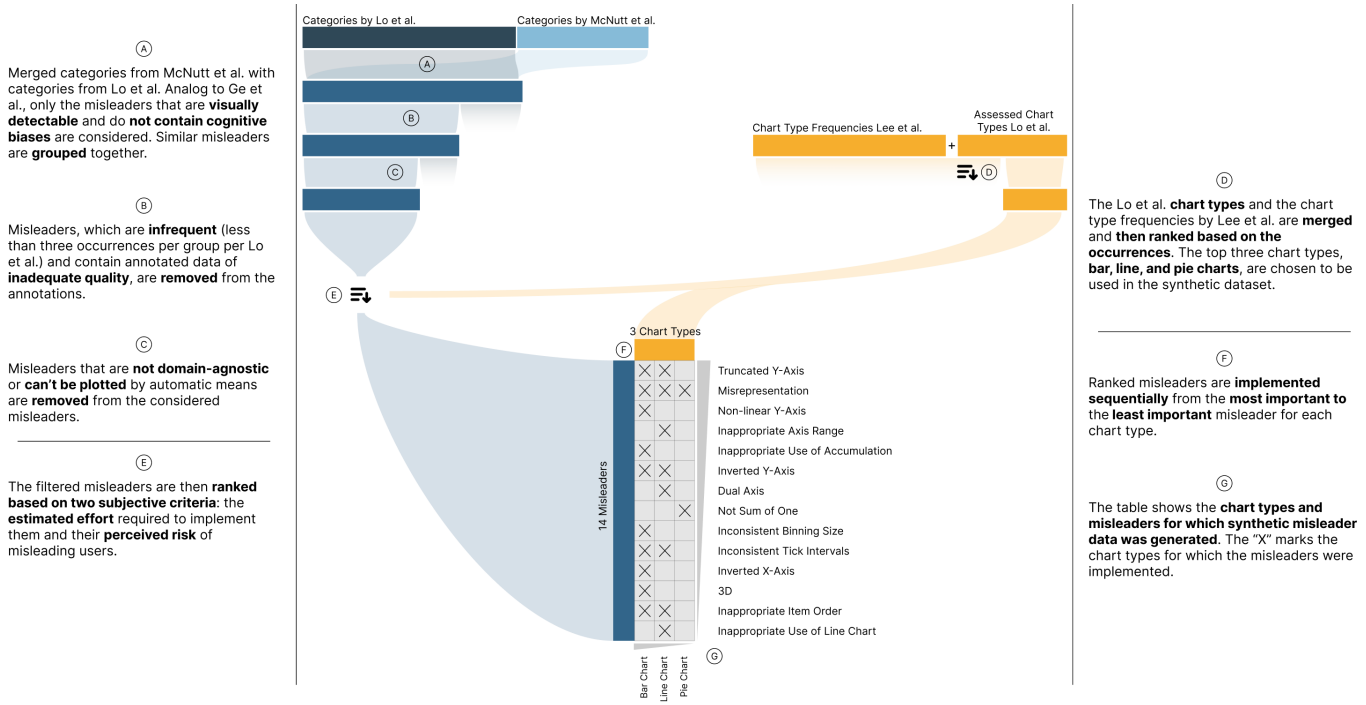


Figure 3.1.: **Overview of the Derivation Process for Chart and Misleader Types in the Synthetic Dataset.** The left section presents the derivation of misleader types included in the dataset, while the right section shows the derivation of chart types used for synthetic generation. The bottom section summarizes the implemented misleader and chart types, indicating the specific chart types to which each misleader is applied.

is assumed that such mistakes are mitigated by the fact that the misleading element is easily recognizable, making viewers aware that interpreting precise values may be unreliable.

Furthermore, misleaders that require domain-specific knowledge to be identified were excluded (Ⓒ in Figure 3.1), as they are difficult to generate automatically without access to external context. For example, the misleaders *dubious data* and *selective data* were removed, as detecting them depends on prior knowledge of the underlying dataset. As a next step, the chart types to be included in the synthetic dataset were derived, as misleader applicability often depends on the chart format. Chart frequencies as reported by Lee et al. [36] and Lo et al. [5] were considered. These chart types were then filtered for those that are straightforward to generate automatically. Based on the filtered frequency rankings, the top three chart types, bar, line, and pie charts, were selected to be included (Ⓓ in Figure 3.1).

The filtered misleaders were then ranked based on two subjective criteria to determine the order of implementation:

- The estimated implementation effort required to automate the misleader creation.
- The perceived risk of the misleader on user interpretation.

The risk associated with each misleader was estimated based on multiple criteria: its frequency of occurrence in real-world visualizations relative to all encountered misleading data visualizations [5], the number of applicable chart types, and a subjective assessment of both the likelihood of causing incorrect reasoning and the impact of its effect on viewer interpretation. This approach ensures that both high-impact but complex

misleaders and simple to implement, but less impactful ones are incorporated in the synthetic dataset (Ⓔ in Figure 3.1). For example, the misleader *misrepresentation* occurs frequently according to Lo et al. [5] and affects all three selected chart types. If unrecognized, the subjectively derived impact is assumed to be large. Ease of implementation is expected to be of medium difficulty. Considering all factors, the misleading factor was ranked second. The resulting ranking defines the implementation order. Misleaders were then systematically implemented for all applicable chart types (Ⓕ in Figure 3.1). An overview of the implemented misleaders and corresponding chart types is presented in 3.1 Ⓖ. Appendix A.1 provides an overview of the misleader types implemented and their definitions.

3.4. Technical Implementation

Choice of Real-World Base Data and Charting Framework
As outlined in the general requirements, the misleading data visualizations created in this work are based on real-world, open-domain datasets. Specifically, this project utilizes data from Our World In Data (OWID) [72] and TabFact [73]. Both datasets are shared under the Creative Commons (CC) BY license [74], making them freely distributable and modifiable. OWID is an open-access platform that publishes data-driven research on global challenges such as health, education, the environment, and economic development [72]. At the time of writing, the platform provided access to 123 publicly available datasets [72]. TabFact is a dataset designed for table-based fact verification, containing 16.000 data tables scraped from Wikipedia. These tables and the OWID datasets form the basis for generating synthetic data visualizations in this work. *Misviz Synthetic* is released under the CC BY license [74]. The `matplotlib` framework [75] is used to generate the data visualizations, as it ensures reproducibility and meets the general requirements for a highly configurable charting interface. In particular, its flexibility is essential to support this work's wide range of chart types and misleader variations.

Plotting Process
The misleading data visualization plotting process is divided into two main steps. The data column types, relevant column combinations, and suitable chart types are determined in the first step. In the second step, the data are passed to misleader plotters, which generate a misleading data visualization if possible for the given input (see Figure 3.2).

Dividing the process into two steps allows flexibility in chart creation, enabling the reuse of intermediate data across different plotting frameworks. For example, instead of using `matplotlib` [75], one could employ other visualization libraries to create additional data for the synthetic misleading data visualizations dataset. Each misleading data visualization contains either one misleading factor or none. Two examples of the *Misviz Synthetic* dataset can be seen in Figure 3.3.

In more detail, the plotting process begins by identifying a natural key to serve as the primary indexing column for the input tables. The objective is to ensure that each numerical value in the table can be meaningfully associated with a unique key. Preference is given to non-numerical and previously identified temporal columns, as these are more likely to convey semantic structure. In contrast, purely numerical fields often lacked contextual indicators that identify them as natural key candidates and were less reliable for uniquely identifying table records. Once a column (or minimal combination of columns) uniquely identifying individual records is found, it is paired with previously identified numerical columns not contained in the natural key. If the natural key consists of multiple columns, all but one are conditioned on specific values to produce meaningful groupings by the remaining variable. This strategy is particularly suited for large-scale datasets from OWID [72], which commonly include multiple metrics reported over time across different countries.

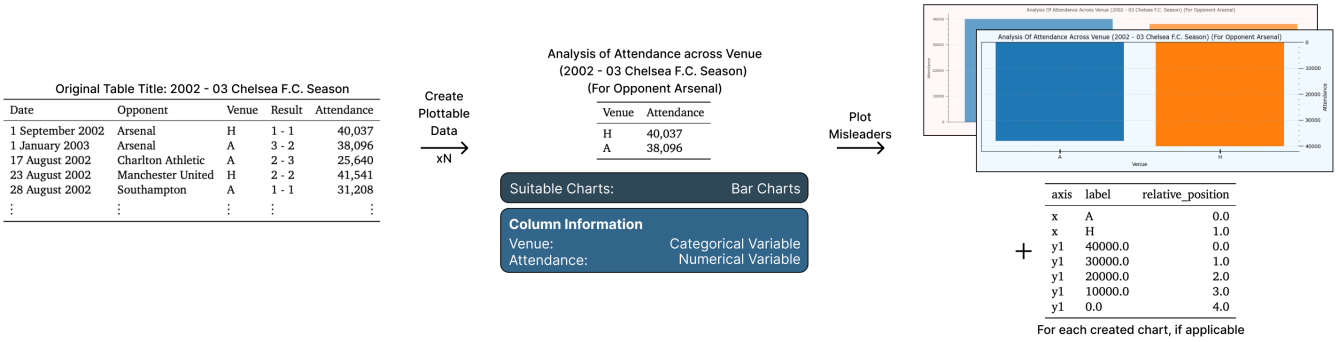


Figure 3.2.: **Illustration of the Misleading Chart Generation Process.** The left panel displays an input table sourced from the TabFact Wikipedia dataset [73]. In the first processing step, the system extracts column data types as well as potential column and value combinations. The result is shown in the center, illustrating one such combination along with a suitable chart title, chart type, and metadata about the column types. In the second step, this configuration is passed to the misleader plotters, that generate misleading and non-misleading visualizations based on the original table. In addition, axis metadata is extracted from coordinate-based data visualizations. The generated visualizations are shown in the right panel. The bar chart in the foreground includes an *inverted y-axis* as a misleading factor, while the chart in the background represents a non-misleading version of the same data. Additionally, axis data for each chart is extracted during the plotting process, if applicable.

As an example, a typical schema may contain key columns such as `Country`, `Year`, and a tracked metric (e.g., `GDP`). If the natural key is `Country + Year`, holding the year constant (e.g., 2012) yields a bar chart comparing the selected metric across countries. Conversely, conditioning on the country (e.g., France) results in a time series showing the evolution of the metric over the years for that specific country, which can be plotted as a bar or line chart. Following data filtering, a chart title is generated using predefined templates that incorporate the original table name and the names of the selected independent and dependent variables. In cases where the original natural key comprises multiple columns, the held constant dimension is also included in the title to preserve contextual clarity. Example titles include: *Original Table Name: GDP per Year (for Country France)* or *Original Table Name: GDP per Country (for Year 2012)*, depending on which key dimension is held constant.

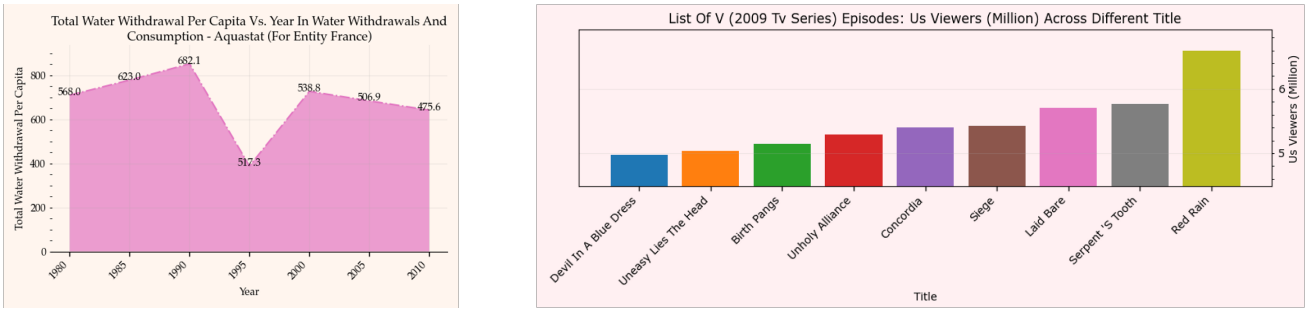


Figure 3.3.: **Two Examples from the *Misviz Synthetic* Dataset.** The left image shows a line chart containing the *misrepresentation* misleader, as the value labels do not correspond to the values when projected onto the y-axis. The right image depicts a bar chart with a *truncated y-axis*, making the value on the right appear disproportionately larger compared to the other values in the chart.

Suitable chart types are selected based on the column types and the number of records. The resulting filtered data, along with additional information on column types, applicable chart types, and chart titles, is then saved as an intermediate step.

In the second process step, each individual extracted data table from the intermediate step is passed to misleader plotters, which introduce the misleading factors into a data visualization. Each misleader plotter filters by chart type and misleader-specific conditions the data must satisfy. The visual appearance of each misleading chart is further diversified through randomized parameters, including background color, axis label positions, and tick styles (see Appendix A.4 for an extensive list). For each generated visualization, both the manipulated data table and the corresponding axis metadata are stored in `.csv` and `.json` formats, respectively. Overall, the final dataset contains 83.038 data visualizations.

3.5. Dataset Metadata

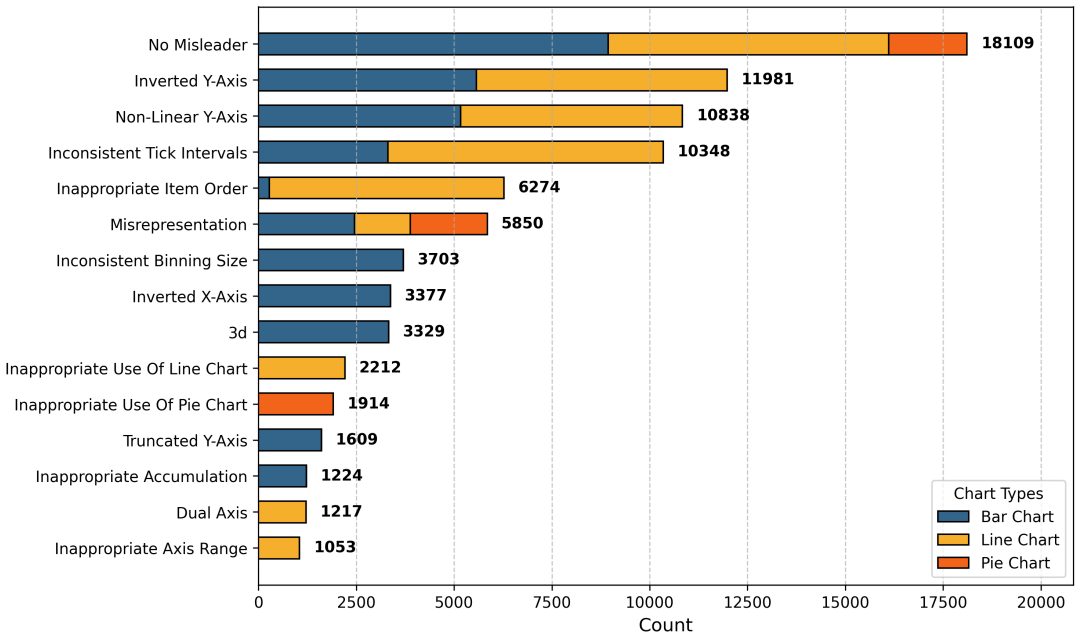


Figure 3.4.: **Distribution of Chart Visualizations per Label.** Counts of misleader types and relative fractions across chart types.

The *Misviz Synthetic* dataset contains 14 different misleading chart types and a category representing charts without misleaders. The most frequent category is *no misleader*, followed by commonly occurring distortions such as *inverted y-axis*, *nonlinear y-axis*, and *inconsistent tick intervals*. Less prevalent misleaders, such as *dual axis* and *inappropriate axis range*, appear in smaller numbers due to their limited applicability across the available tables. Bar, line, and pie charts are represented across the dataset. However, certain misleaders are restricted to specific chart types (e.g., *inappropriate use of pie chart* and *inappropriate use of line chart*). An in-depth overview of the distribution of misleaders and chart types can be seen in Figure 3.4.

The *Misviz Synthetic* dataset is divided into training, validation, and test splits to facilitate training, evaluation, and hyperparameter tuning. The training set contains 77.038 samples. An additional subset of 15.000 examples (*Train Small*), drawn from the training set, is provided for faster prototyping. The validation and

test sets include 2.000 and 4.000 samples, respectively, leading to a total of 83.038 chart images. A stratified split ensures a similar distribution across all subsets. For additional information on the data plotting process, conditions under which misleaders are introduced, and further data visualization examples, see Appendix A.3 and A.2.

3.6. Limitations of the *Misviz Synthetic* Dataset

Despite efforts to ensure comprehensive coverage of misleading data visualizations, several limitations remain. Due to the complexity of introducing multiple misleader types, the dataset consists only of visualizations containing a single misleader type per chart. In contrast, real-world data visualizations often contain multiple misleading factors per chart.

Additionally, pie charts are underrepresented, accounting for only about 7% of the total dataset. This is primarily due to the difficulty of automatically identifying semantically valid part-of-whole relationships in structured tables. A rule-based extraction method was used to identify candidate columns for pie chart generation. To address this issue, synthetic data usage can be explored, or additional real-world data suitable for pie charts must be collected.

Not all misleading visualization types could be implemented to their full extent. For instance, the *inappropriate axis range* misleader is currently limited to cases where the axis is too narrow. At the same time, overly broad axes, although potentially misleading, were not addressed because reliably identifying them requires contextual reasoning that is difficult to automate.

The dataset is also constrained by the capabilities of the `matplotlib` library for chart rendering. Some chart types, such as 3D line charts or 3D pie charts, could not be generated due to library limitations. Furthermore, several chart types and variants commonly used in practice, such as horizontal bar charts, stacked bar charts, donut charts, and geographic maps, were not included in the current version of the dataset. However, the intermediate output from the initial data processing already includes structured data suitable for these chart types, indicating that the dataset could be extended in this direction through future work. In addition, expanding the dataset to include charts generated with libraries beyond `matplotlib` could enhance the visual diversity of the synthetic data and better reflect the variety of styles observed in real-world visualizations.

Another limitation relates to chart titles. Titles were generated using predefined templates incorporating column names and filters derived from the natural key approach. While this ensures consistent formatting, the resulting titles are often verbose and complicated, especially when original table or column names include parenthetical characters. For instance, a title such as “Share of employment in the financial sector (GGDC, 2017): Share of workers in financial sector by Year (for Entity South Korea)” lacks natural readability and fluency. Future work could explore more flexible and adaptive title generation, for example, through the use of LLM to produce more concise and readable titles. Because the underlying data is predominantly in English, most of the generated charts and title templates are likewise in English, constituting a further limitation of the dataset.

While the synthetic dataset provides a more balanced representation of the included misleaders which supports practical training and systematic evaluation, it does not reflect the actual frequency distribution of misleading chart types.

4. Experiments and Results

This chapter presents a series of experiments that evaluate the effectiveness and generalization capability of models trained to detect misleading data visualizations. The experimental pipeline consists of three sequential stages, each building on insights from the previous one.

In **Stage 1**, an ablation study is conducted to assess the contribution of different input features, namely the chart image, ground truth axis metadata, and the ground truth underlying data table, to the performance of a classification model. All models in this stage are trained on the training split of the *Misviz Synthetic* dataset.

Based on these findings, the two best-performing vision encoder models are selected for further evaluation in **Stage 2**. Here, the selected configurations are retrained on the small training subset of the synthetic dataset. In contrast to Stage 1, axis metadata is no longer taken from the ground truth data but is extracted directly from the chart image using *Axis-DePlot*, a custom-trained *DePlot*-based [42] axis extractor, approximating a real-world scenario where the raw image is the sole input source. The model is evaluated using the test set of *Misviz Synthetic*.

In **Stage 3**, the trained models from Stage 2 are evaluated on the real-world *Misviz* test set to assess their ability to generalize beyond synthetic data and perform in realistic, previously unseen scenarios.

To summarize, the experiment stages are structured as follows:

- **Stage 1:** Ablation Study on Input Features
- **Stage 2:** Training under Realistic Input Conditions
- **Stage 3:** Evaluation on Real-World Charts

Together, these experiments form a stepwise evaluation pipeline that investigates the role of input features in model performance **RQ1**, the capacity to train models using realistically available inputs **RQ2**, and the potential for generalization from synthetic to real-world data **RQ3**.

4.1. Ablation Study on Input Features

The first stage of the experimental pipeline addresses **RQ1**, investigating the contribution of different input features to the task of detecting misleading data visualizations. Specifically, this ablation study evaluates how combinations of chart image features, axis metadata, and underlying data table representations can affect classification performance in detecting misleading data visualizations. By systematically enabling and disabling individual input components, the goal is to determine which features are most informative for the detection task. For this purpose, the *Misviz Synthetic* dataset is utilized to train and evaluate the models. The ground truth axis metadata and underlying data included in the synthetic dataset are utilized in the experiments.

This stage provides the foundation for subsequent experiments by identifying the best-performing model configurations for Stage 2 training.

4.1.1. Classifier Model Architecture

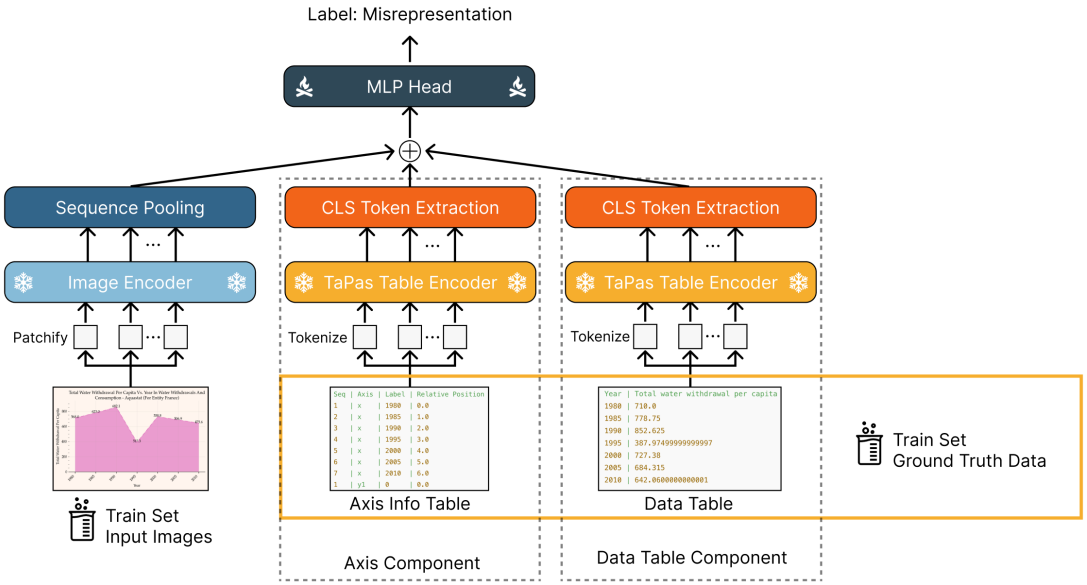


Figure 4.1.: **Ablation Study Training Architecture.** A vision model encodes the chart image, while optional axis and data components are encoded using TaPas [76]. The resulting embeddings are concatenated and passed to a classification head for prediction. Snowflakes indicate that the component is frozen during training while the fire icon indicates that the component is actively trained.

Model Name	Vision Encoder Architecture	Max Input Size	#Parameters
CLIP [77]	ViT-L/14 [53]	336x336	304M
SigLIP [78]	SoViT-400m/14 [54]	384x384	428M
DePlot [42]	ViT-B/16 [53]	-	282M
UniChart (ChartQA-960) [62]	Swin Transformer [55]	960x960	201M
TinyChart-3B-768 [65]	SoViT-400m/14 [54] + ToMe [79]	768x768	428M

Table 4.1.: **Models Used in Vision Encoder Evaluation.** The table includes model architectures, input resolution constraints, and parameter counts. The *DePlot* encoder differs from the other vision encoders in that its maximum image input size is limited by the number of input image patch tokens rather than a fixed resolution.

All ablation study configurations use a fixed architecture with a multi-class classification head. The image input is always included, while the axis and data components are added in different combinations (see Figure 4.1). A vision encoder extracts image representations from the chart image and is included in all model configurations. Axis and table inputs are optionally added, depending on the specific configuration, and are exclusively encoded using the TaPas model `tapas-large-finetuned-wtq` [76], with the `[CLS]` token serving as their output representation. The outputs from all components are concatenated and passed to a

classification head implemented as a Multi-Layer Perceptron (MLP). Five vision encoders are evaluated across all configurations, each using mean sequence pooling to obtain a fixed-length image representation. To ensure broad coverage of vision models relevant to chart understanding, both general-purpose and chart-specific encoders are considered. Among the general-purpose models used in this work are *CLIP* [77] and *SigLIP* [78], which are widely adopted in MLLMs for chart reasoning tasks [60, 64, 66, 80]. In addition, chart-specialized encoders, such as those used in *DePlot* [42], *UniChart* [62], and *TinyChart* [65], are included to represent models fine-tuned or explicitly developed for chart-related data. *TinyChart* applies Token Merging (ToMe), which dynamically merges similar tokens during inference to downsize input size and speed up transformer-based models, with minor impact on accuracy [79]. Combinations of multiple encoders or vision towers are not considered in this work [64, 65, 67, 81]. All used vision encoder models and additional information can be seen in Table 4.1.

4.1.2. Model Training

All vision and table encoders remain frozen during training. Only the MLP-based classification head is updated. For each of the five vision encoders, four input configurations are tested (see Section 4.1), differing in the inclusion of axis metadata and the underlying chart data:

- Image only
- Image + Axis metadata
- Image + Chart data
- Image + Axis metadata + Chart data

It is important to note that the axis metadata and underlying table data are not extracted from the chart image. Instead, the ground truth data from the synthetic dataset is used. The training data is augmented with variations in rotation and perspective. All models were trained over 300 epochs with early stopping enabled. A batch size of 256 and the Adam optimizer [82] with a learning rate of 5×10^{-5} was used for training. The loss was weighted by the misleader frequency. The trained MLP has the input dimensions of the concatenated input features, a hidden dimension of 1.024, and an output dimension equal the number of the misleader labels. Each model configuration was trained with three different random seeds, and for each, the best-performing model on the validation set of *Misviz Synthetic* was chosen for evaluation.

4.1.3. Evaluation

Model performance is evaluated on the test split of the *Misviz Synthetic* dataset. To assess the effectiveness of each configuration in detecting misleading data visualizations, the macro-averaged F_1 score is used as the primary evaluation metric. The F_1 score, computed as the harmonic mean of precision and recall, provides a balanced measure of a model's performance by accounting for both false positives and false negatives. In the multi-class setting, macro averaging computes the F_1 score independently for each class. Then it takes the unweighted mean, treating all classes equally, regardless of their frequency in the dataset. This choice is relevant in the context of this work, as not all misleader types are represented in the same quantity in *Misviz Synthetic*. Macro F_1 therefore ensures that performance on minority classes, such as *dual axis* or *inappropriate axis range*, is not overshadowed by more common classes. The F_1 score for a single class is given by:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.1}$$

Precision and recall are defined as:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{4.2}$$

Consequently, the macro-averaged F_1 score is calculated as follows:

$$F_1^{\text{macro}} = \frac{1}{C} \sum_{i=1}^{C} F_1^{(i)} \tag{4.3}$$

where C denotes the number of classes and $F_1^{(i)}$ is the F_1 score of the i-th class.

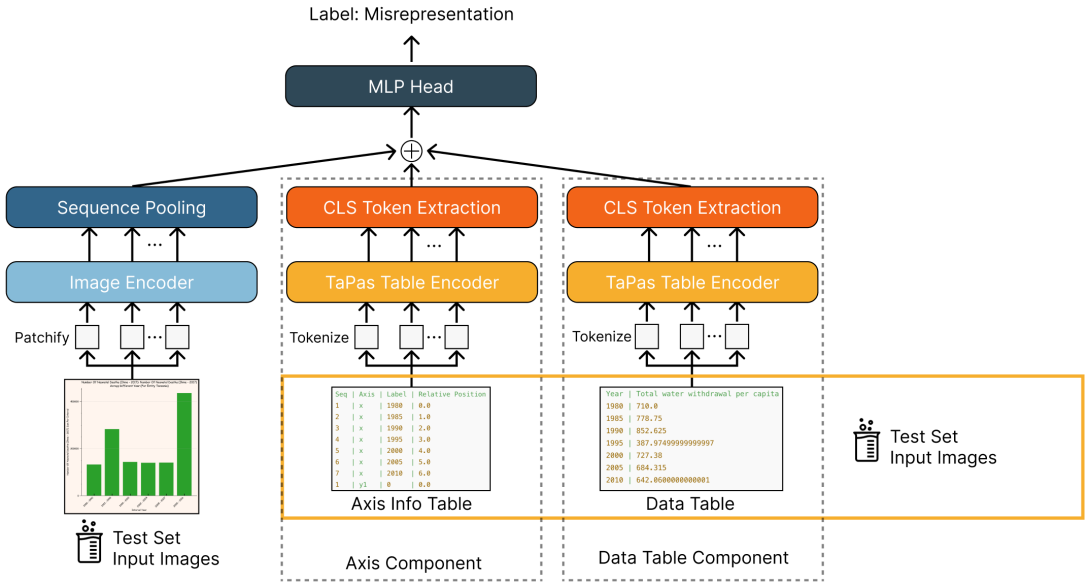


Figure 4.2.: **Ablation Study Inference Architecture.** The architecture mirrors the training setup: A vision model encodes the chart image, and optional axis and data components are embedded using TaPas [76]. Embeddings are concatenated and passed to a classification head for prediction. During evaluation, inputs are drawn from the *Misviz Synthetic* test split.

Similar to the training phase, the ground truth underlying data and ground truth axis metadata are used (see Figure 4.2). In the following sections, all reported F_1 values refer to the macro-averaged F_1 score unless stated otherwise. In addition, the standard deviation of the F_1^{macro} score is reported for each model configuration, based on the best-performing model from each of the three random seeds, to account for training variability.

4.1.4. Hypotheses

This experiment explores how different input features and vision encoder model types might affect the detection of misleading data visualizations. It is hypothesized that models incorporating underlying chart data are expected to perform better, benefiting from direct access to the original numerical values. Furthermore, models that include axis metadata are expected to outperform those relying solely on the underlying data or visual representation, as axis elements are often directly manipulated in misleading visualizations (e.g. *truncated y-axis*, *inconsistent tick intervals*). It is also hypothesized that chart-specific vision encoders, designed or fine-tuned for structured chart data, will outperform general-purpose encoders due to their specialized training on domain-relevant patterns and semantics.

4.1.5. Results

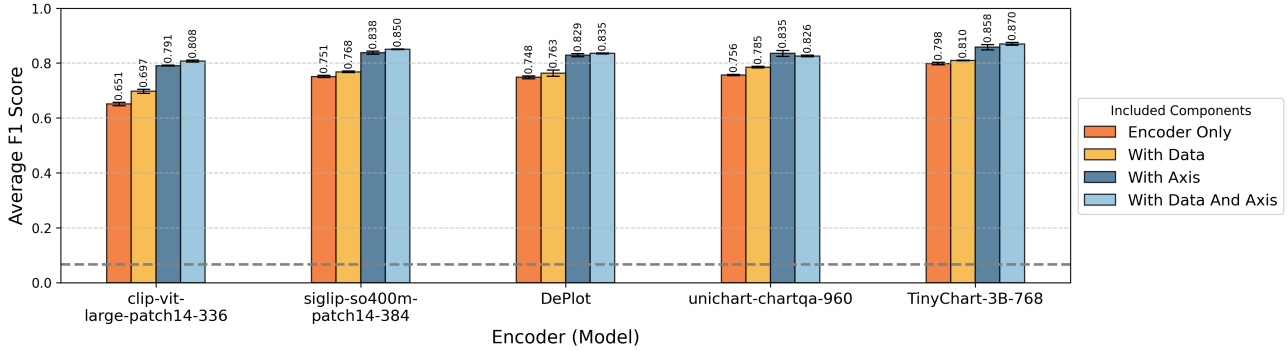


Figure 4.3.: **Ablation Study Results.** The x-axis shows the base vision encoder models used, and the y-axis shows the macro-averaged F_1 score. Each bar represents the macro-averaged F_1 over three runs with different random seeds, with the standard deviations shown on top of the bars. The gray dashed line represents the random baseline.

The results show improvements in F_1 performance when axis metadata is incorporated as input to the classifier head, while the inclusion of underlying chart data yields only marginal improvements. Models based on the *TinyChart* vision encoder [65] achieve the best performance across all configurations, with all four trained variants surpassing other encoders in their respective configurations. The highest-performing model overall uses the *TinyChart* vision encoder with integrated data and axis metadata. It achieves a best single-run F_1^{macro} of 0.876, with an average F_1^{macro} of 0.870 across three independent runs. The second-best model uses the *TinyChart* vision encoder with axis metadata and image representation, achieving a best single-run F_1^{macro} of 0.867 and an average of 0.858 across three independent runs. In contrast, *CLIP*-based models [77] exhibit the lowest performance among all configurations tested.

To better understand where these improvements occur, a per-class analysis was conducted. Comparing the per-class F_1 score differences between the best encoder-only model (*TinyChart*) and the best model using both encoder output and axis metadata (*TinyChart* + ground truth axis metadata) shows that the addition of the axis metadata leads to improvements across almost all misleader types. The most substantial gains were observed for misleaders that directly manipulate axis properties, with the most significant improvements observed for *inappropriate axis range* (+0.187), *nonlinear y-axis* (+0.080), *inconsistent intervals* (+0.135), *inverted x-axis* (+0.098), *inverted y-axis* (+0.052), *inappropriate item order* (+0.066), and *no misleader*

(+0.093). These results highlight consistent gains from incorporating axis metadata into the classification model.

While models relying solely on the vision encoder output already achieve strong performance, they perform worse on misleader types that require information on the axis. The worst class-based F_1 scores can be observed for the misleader classes *truncated y-axis*, *inappropriate axis range*, *inconsistent tick intervals*, and *nonlinear y-axis*. All classes show considerable improvements once the axis metadata is included.

Chart-specific vision encoders do not show a significant advantage over general-purpose encoders. The *SigLIP* vision encoder [78] consistently outperforms the vision encoder of both the *DePlot* [42] and *Unichart* [62] model, despite both being designed explicitly for chart domain tasks.

4.1.6. Interim Conclusion

The results of the ablation study provide insights into **RQ1**: *Which feature of a chart (the image itself, the axis metadata, or the underlying data table) contributes to the detection of misleading factors?* The findings show that while vision encoders offer a strong baseline for detecting misleading elements, they often lack explicit structural information, particularly axis-related cues. This becomes evident in misleader types involving axis manipulation, such as *inappropriate axis range* and *inconsistent tick intervals*, where adding axis metadata consistently improved model performance across all configurations. In contrast, the underlying data table offered limited benefits once axis metadata was included, suggesting that the visual or axis features already capture most relevant signals for the detection task.

The results also challenge the initial assumption that chart-specific vision encoders universally outperform general-purpose vision encoders. While the encoders of chart-specialized models such as *TinyChart* [65] achieved the best results overall, general-purpose encoders like *SigLIP* [78] outperformed the vision encoders of chart-focused models such as *DePlot* [42] and *UniChart* [62] in several settings. This suggests that the pretraining of the general-purpose models on large-scale visual data may already impart a degree of awareness of chart structure.

That no improvement can be seen for the misleader *truncated y-axis* when adding the axis information can be attributed to the fact that approximately 50% of the bar charts in the dataset do not have a y-axis plotted, but solely the x-axis. As a result, models must rely on visual cues, such as the proportional differences between bars, rather than explicit axis information to detect truncation. In contrast, the *inappropriate axis range* misleader, which in the context of the synthetic dataset only affects line charts where y-axes are always present, shows a substantial performance increase when the axis metadata is included. However, this also highlights that the image representations alone lack structural knowledge of the chart in cases where the y-axis is absent, limiting the model's ability to detect axis-based misleaders without the y-axis information.

The top-performing models, which combine the *TinyChart* encoder with both axis metadata and chart data, achieved a macro-average F_1 score of 0.870. A nearly identical performance was observed when using only the axis metadata and image features ($F_1^{\text{macro}} = 0.858$), reinforcing the importance of explicitly incorporating axis features when addressing misleading chart detection tasks.

4.2. Training under Realistic Input Conditions

Based on the results of the ablation study, model variants incorporating both vision encoder image representations and axis metadata are selected for further training, as including axis metadata substantially improved the detection of misleading data visualizations. In this stage, the classifier is extended to extract axis information directly from the chart image, replacing the ground truth axis metadata employed in the previous stage. To enable this, a new model called *Axis-DePlot* is developed by fine-tuning *DePlot* [42] for axis metadata extraction. The two best-performing vision encoders from the previous stage, *TinyChart* and *UniChart*, are used. In addition, the vision encoder component of *Axis-DePlot* is evaluated in the same setup to compare its learned image representations. To assess the effect of the extracted axis metadata as an additional component in the classifier, a baseline model using only the visual encoder is included, allowing for comparison in cases where axis extraction may introduce errors. As before, the *Misviz Synthetic* dataset is used to train the resulting models, with the difference being, that not the ground truth but the extracted axis metadata is used. Configurations incorporating underlying chart data are omitted at this stage, as they comparatively yielded only marginal improvements in the ablation study.

4.2.1. The *Axis-DePlot* Axis Extractor

Misviz Synth Input Image

Axis Info Table

Figure 4.4.: **Depiction of the *Axis-DePlot* Axis Metadata Extraction.** The model extracts axis metadata from chart images to replace ground truth information used in previous training stages. Only a subset of the axis metadata is shown.

Unlike in the ablation study, axis metadata is unavailable at inference time for the task of detecting misleading data visualizations. Therefore, the axis information must be extracted automatically from the chart image. A fine-tuned variant of the *DePlot* model [42] is employed to address this. Initially developed for chart-to-table data extraction, *DePlot* is a lightweight model well-suited for axis extraction due to its prior training on chart-related data. *DePlot* is initialized from *MatCha* [61], a vision-language model designed for mathematical reasoning over visual inputs, which itself is initialized from *Pix2Struct* [83], a model trained on image-text pairs from infographics and websites to learn visual grounding for structured outputs. Building on this foundation, a new model called *Axis-DePlot* is trained to extract axis tick labels and their relative positions directly from chart images.

***Axis-DePlot* Table Format**

In *DePlot*, the output is a markdown-formatted text sequence, where "|" separates individual cells and "\n" separates rows [42]. For axis extraction, the output of the base *DePlot* model is modified. The axis metadata table produced by the fine-tuned *DePlot* model consists of a structured sequence of axis tick entries, where each row represents a single tick mark extracted from the chart image. The table includes the following four columns:

- **Seq:** A sequential index indicating the order in which tick marks appear along a given axis. Tick indices are read out from bottom to top for vertical axes and from left to right for horizontal axes.
- **Axis:** A categorical label specifying the axis to which the tick belongs (e.g., `x`, `y1`, or `y2` in the case of dual axis charts).
- **Label:** The textual or numerical value associated with the tick, extracted directly from the chart (e.g., `1950`, `50000` or `France`).
- **Relative Position:** A normalized float indicating the tick's position along the axis. The spacing between the first two detected ticks (from bottom-to-top for y-axes and left-to-right for x-axes) is set to 1.0, and all subsequent positions are expressed relative to this reference distance.

A partial example of the axis extraction output table format is shown in Figure 4.4.

Training of *Axis-DePlot*

The axis metadata extraction model was trained on the large training split of the *Misviz Synthetic* dataset. To improve the model's sensitivity to varying tick intervals and step sizes, an additional 3.500 synthetic chart images were generated using `matplotlib` [75], designed to introduce diverse axis spacings. The model was initialized from the pretrained *DePlot* checkpoint.

Training was conducted on a distributed setup with two NVIDIA H100 GPUs and ran for four epochs using a batch size of 8 and a target sequence length of 1024 tokens, twice the default output length of 512 tokens from the original *DePlot* configuration. The Low-Rank Adapter (LoRA) [84] method was applied to the query and value projection layers with a rank of 16 to support memory-efficient training.

To improve generalization, input images were augmented with random rotations and perspective transformations. Although the *Pix2Struct* model [83], on which *DePlot* is based on, was trained using the AdaFactor optimizer [85] with weight decay (10^{-5}), linear warm-up to a peak learning rate of 0.01, and cosine decay, this configuration led to divergence during fine-tuning of *Axis-DePlot*. As a result, the Adam optimizer [82] with a constant learning rate of 5×10^{-5} was used instead, resulting in stable convergence during fine-tuning.

Evaluation of *Axis-DePlot*

To evaluate the similarity between the predicted table output of the *Axis-DePlot* model and the ground-truth tables, a metric derived from the *Relative Mapping Similarity (RMS)*, as proposed by Liu et al. [42], is used. RMS treats tables as unordered collections of mappings from row and column headers to corresponding values. In RMS, each entry in the predicted table $P = \{p_i\}_{i=1}^{N}$ and target table $T = \{t_j\}_{j=1}^{M}$ is represented as a triplet. For the axis extraction task, the tuple is expanded to contain four elements instead of three, to reflect the new table structure:

$$p_i = (p_i^{\text{seq}}, p_i^{\text{axis}}, p_i^{\text{label}}, p_i^{\text{rel\_dist}}), \quad t_j = (t_j^{\text{seq}}, t_j^{\text{axis}}, t_j^{\text{label}}, t_j^{\text{rel\_dist}}),$$

where p_i^{seq}, t_j^{seq} denote the sequence number of the record, p_i^{axis}, t_j^{axis} the axis on which the tick can be found, $p_i^{\text{label}}, t_j^{\text{label}}$ the respective axis labels, and the relative distance $p_i^{\text{rel\_dist}}, t_j^{\text{rel\_dist}}$.

Similarities of cell content with text is measured using the normalized Levenshtein distance, denoted as $\text{NL}_\tau(a, b) = \min(1, \text{NL}(a, b))$, where a and b are strings formed by concatenating row and column headers. Numerical similarity is computed using a relative distance with threshold θ. Distances above θ are set to the maximum of 1:

$$D_\theta(p, t) = \min\left(1, \frac{||p - t||}{||t||}\right).$$

The combined similarity between entries p_i and t_j is defined as:

$$D_{\tau,\theta}(p_i, t_j) = (1 - \text{NL}_\tau(p_i^{\text{seq}} || p_i^{\text{axis}}, t_j^{\text{seq}} || t_j^{\text{axis}})) \cdot (1 - (\alpha \cdot S(p_i^{\text{label}}, t_j^{\text{label}}) + (1 - \alpha) \cdot D_\theta(p_i^{\text{rel\_dist}}, t_j^{\text{rel\_dist}}))),$$

with $S(p_i^{\text{label}}, t_i^{\text{label}})$ defined as follows:

$$S(p_i^{\text{label}}, t_i^{\text{label}}) = \begin{cases} 1_{[p_i^{\text{label}} \neq t_j^{\text{label}}]} & \text{if } p_i^{\text{label}} \text{ is numeric} \\ \text{NL}_\tau(p_i^{\text{label}}, t_j^{\text{label}}) & \text{if } p_i^{\text{label}} \text{ is a string} \end{cases}$$

Here, string concatenation is denoted by $||$. The similarity score approaches 1 when the predicted and ground-truth rows are highly similar, and 0 otherwise. When comparing numerical labels, exact matches are required. Even small deviations can prevent detection of misleaders. For example, in misleaders such as *inappropriate item order* or *inconsistent intervals*, a single misread character can lead to an incorrect extraction. If numeric labels do not match exactly, the prediction is considered incorrect, and the similarity value is set to 1, penalizing the score.

In contrast, relative distances between ticks are evaluated more leniently, as they are intended to capture structural properties such as varying step sizes. The weighting factor $\alpha = 0.8$ is chosen to emphasize the correct extraction of labels over precise relative distance between ticks.

Following the RMS approach, a similarity matrix of dimensions $N \times M$ is constructed using the cost function $(1 - \text{NL}_\tau(p_i^{\text{seq}} || p_i^{\text{axis}}, t_j^{\text{seq}} || t_j^{\text{axis}}))$. The optimal assignment between predicted and target entries is then computed using minimum cost matching, resulting in an assignment matrix $X \in \mathbb{R}^{N \times M}$. In this adaptation, the *Seq* and *Axis* columns are used as structural identifiers in place of the row/column headers in standard RMS. This matching approach provides robustness to missing or extra entries in the predicted axis metadata output.

Precision and recall scores are then computed as:

$$\text{RMS}_{\text{precision}} = 1 - \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} X_{ij} \, D_{\tau,\theta}(p_i, t_j),$$

$$\text{RMS}_{\text{recall}} = 1 - \frac{1}{M} \sum_{i=1}^{N} \sum_{j=1}^{M} X_{ij} \, D_{\tau,\theta}(p_i, t_j),$$

And accordingly, the F_1 score can be computed:

$$\text{RMS}_{F_1} = \frac{2 \cdot \text{RMS}_{\text{precision}} \cdot \text{RMS}_{\text{recall}}}{\text{RMS}_{\text{precision}} + \text{RMS}_{\text{recall}}}.$$

A relative distance threshold of $\theta = 0.2$ is used to allow for small deviations in tick position predictions. For string comparisons, the normalized Levenshtein distance with a threshold of $\tau = 0.2$ is applied to tolerate minor label variations. The test set of *Misviz Synthetic* is used to evaluate the axis extraction performance of *Axis-DePlot*.

Fine-tuning Results

The model achieves an average modified RMS_{F_1} score of 90.27%, with a standard deviation of 26.52 percentage points. The main source of variation lies in the predictions for pie charts, where the model often hallucinates values instead of producing no output. This behavior is likely linked to the imbalance in the training data, pie charts account for only around 7%, as the original *DePlot* model was not explicitly trained to suppress

predictions for pie charts. However, the performance on line and bar charts is consistently high. The average inference time per image on the synthetic dataset is approximately 12 seconds if run on a NVIDIA A100 GPU with 40GB RAM, depending on the amount of axis metadata that needs to be extracted.

Revisiting the Ablation Study with *Axis-DePlot*

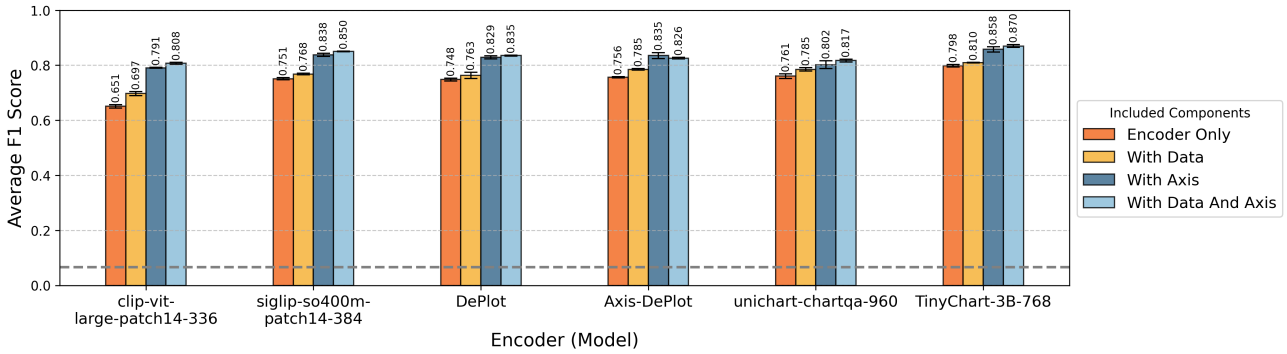


Figure 4.5.: **Ablation Study Revisited Results.** Comparison of ablation study results using the vision encoder from *Axis-DePlot* against the original five models. The *Axis-DePlot* encoder achieves marginal improvements in macro-averaged F_1 score across all but one input configuration. The gray dashed line represents the random baseline.

To see how well the vision encoder chart representations of *Axis-DePlot* capture relevant information for classifying misleading data visualizations, the ablation study was repeated using the fine-tuned vision encoder from *Axis-DePlot*. The results indicate that *Axis-DePlot* achieves marginally improved performance over the original *DePlot* model across three out of four configurations (see Figure 4.5). Models using the *Axis-DePlot* vision encoder also show improved performance when provided with explicit axis metadata.

4.2.2. Classifier Model Architecture

The model employed to detect misleading data visualizations using only the chart image as input leverages a similar architecture as seen in the ablation study. However, instead of relying on ground-truth axis metadata, the *Axis-DePlot* model is employed to extract the axis information from the input chart images (see Figure 4.6).

4.2.3. Training of the Classifier

The classifier is trained on the small training split of the *Misviz Synthetic* dataset to reduce the computational overhead associated with axis metadata inference. The training configuration follows that of the ablation study, with the exception of the maximum number of epochs, which is increased to 500 to account for the reduced training set size.

4.2.4. Evaluation

The classifiers based on the two best vision encoders from the ablation study and the *Axis-DePlot* encoder are evaluated on the test split of the *Misviz Synthetic* dataset. The best-performing model for each configuration is selected based on the validation set. For comparison, this work evaluates two open-source SOTA MLLMs

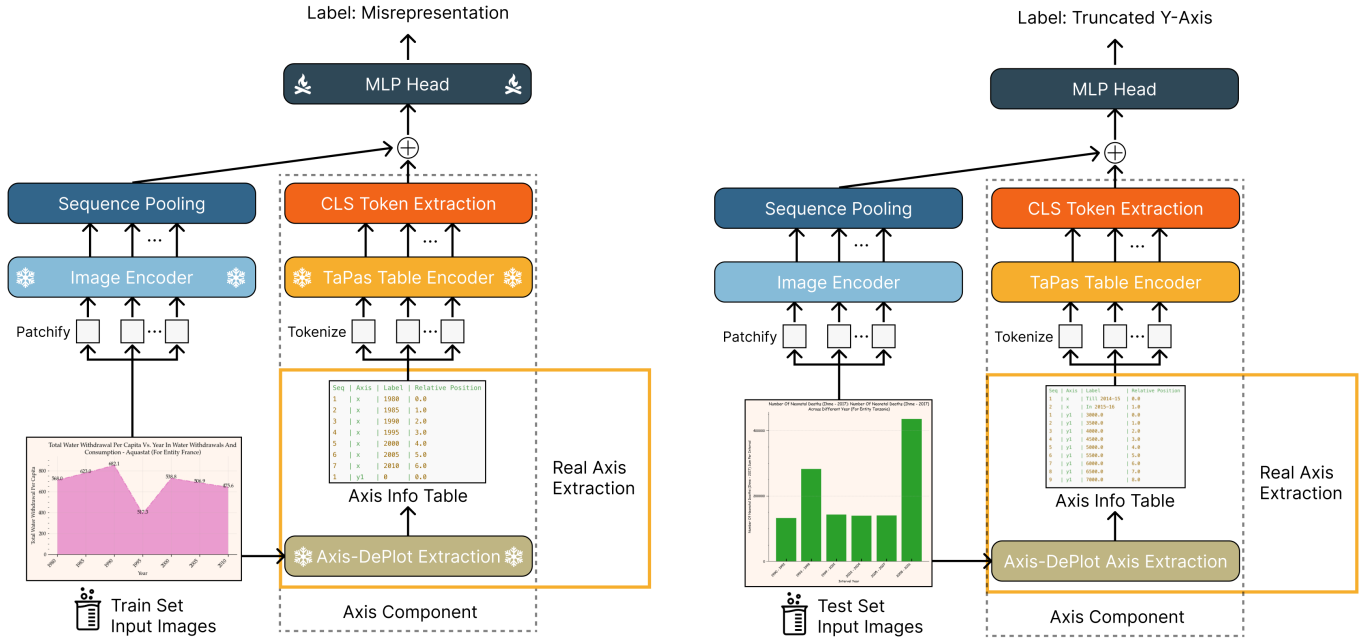


Figure 4.6.: **Training and Inference Architecture.** The left diagram shows the architecture used during training, while the right illustrates the architecture at inference time. As before, the vision model encodes the chart image, and optional axis and data components are embedded using TaPas [76]. Compared to the ablation study, the axis data is now extracted from the input image using *Axis-DePlot*. All outputs are concatenated and passed to a classification head for prediction.

Qwen2.5-VL-7B-Instruct[13] and *InternVL2.5-8B*[14], both of which are among the best-performing publicly available models in the 7–10B parameter range. These models are prompted in a zero-shot setting to classify the same test set. While larger multimodal models such as *GPT-4V* [70] or *Claude Sonnet 3.7* [86] may offer higher performance, they are closed-source and were excluded to ensure a fair and reproducible comparison.

Each MLLM receives a description of all possible misleaders and must assign a single label to each chart image (for full prompt text, see Appendix B.1). No CoT prompting is used. Additionally, a random classifier is included as a baseline. All models are evaluated using the macro-averaged F_1 score (F_1^{macro}) to ensure consistent performance comparison.

4.2.5. Hypotheses

The inclusion of extracted axis metadata is expected to yield a performance improvement over models that rely solely on the encoded image features. Additionally, the models trained on the small training split of *Misviz Synthetic* are hypothesized to outperform the MLLM baselines. The MLLMs are expected to outperform the random baseline.

4.2.6. Results

The results in Figure 4.7 show that models trained on the synthetic data outperform the MLLMs by a large margin. The F_1^{macro} scores for both MLLMs are only slightly above the random baseline. All models trained on

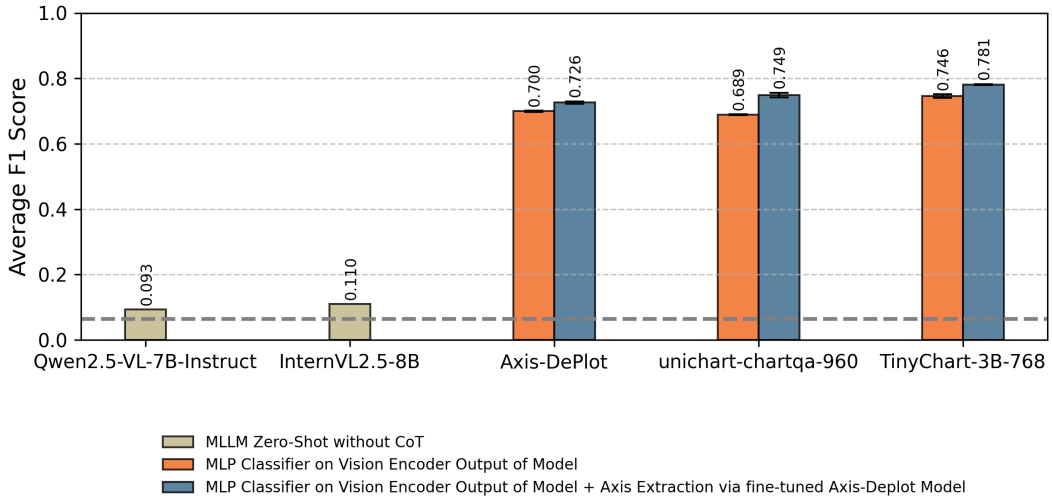


Figure 4.7.: **Macro-Averaged F_1 Score Comparison across Models and Configurations on the *Misviz Synthetic* Test Set.** The figure shows the mean performance on the test set of the *Misviz Synthetic* dataset, with macro-averaged F_1 scores computed over three independent runs. The gray dashed line represents the random baseline.

the small training split of the synthetic data and utilizing the *Axis-DePlot* axis extractor outperform the vision-encoder-only configurations. The best-performing model combines the *TinyChart* vision encoder with extracted axis metadata and achieves a F_1^{macro} score of 0.782, with an average of 0.871 across three independent runs. A breakdown of class-wise F_1 scores (see Appendix Table B.2) indicates high performance on visually distinct misleaders, including *3D* ($F_1 = 1.000$), *inverted x-axis* ($F_1 = 0.910$), *dual axis* ($F_1 = 0.983$), and *inappropriate use of line chart* ($F_1 = 0.976$). In contrast, lower scores were observed with *truncated y-axis* ($F_1 = 0.559$) and *inappropriate use of pie chart* ($F_1 = 0.526$). All 15 classes achieved non-zero performance, including subtler misleaders like *nonlinear y-axis* ($F_1 = 0.798$) and *inappropriate axis range* ($F_1 = 0.792$), indicating robust detection across a diverse range of misleaders.

4.2.7. Interim Conclusion

The results of this experiment support the hypotheses and provide an affirmative answer to **RQ2**: *Can a trained model detect misleading data visualizations?* Classifiers trained on the synthetic dataset outperform the evaluated MLLMs baselines, which achieve F_1^{macro} scores only marginally above the random baseline. This suggests that models explicitly trained for the task, using labeled synthetic data, are better suited for detecting misleading visualizations in this setting than general-purpose MLLMs applied in a zero-shot fashion. Across all configurations, models that incorporate extracted axis metadata consistently outperform their vision-encoder-only counterparts, highlighting the relevance of axis metadata for the classification task. This trend is reflected in the best-performing model, which combines the *TinyChart* vision encoder with axis metadata extracted via *Axis-DePlot*, achieving an average F_1^{macro} score of 0.782. Class-wise results indicate strong performance on visually distinctive misleaders such as *3D*, *inverted x-axis*, and *dual axis*, while lower scores on categories such as *truncated y-axis* and *inappropriate use of pie chart* suggest that more subtle or numerically driven misleaders remain more challenging. Overall, these findings indicate that trained models

leveraging vision encoders can detect a diverse range of misleaders in data visualizations under controlled conditions, particularly when supported by axis metadata extraction.

4.3. Evaluating Generalization to Real-World Misleading Data Visualizations

The classifiers trained on the small synthetic train split in Stage 2 (Section 4.2) are now evaluated on real-world misleading data visualizations to assess their ability to generalize beyond the synthetic data. The best-performing checkpoint for each configuration and random seed is selected individually based on performance on the *Misviz* validation set. The classifier architecture remains identical to the setup described in Figure 4.6.

4.3.1. Evaluation

To assess performance in real-world scenarios, the models are evaluated on the test split of the real-world *Misviz* dataset instead of the *Misviz Synthetic* test set. The *Misviz* dataset provides an extensive collection of annotated real-world charts that share a subset of labels and chart types with the synthetic dataset. This partial overlap enables an assessment of model generalization from synthetic to real-world data under realistic conditions.

Model performance is measured using the F_1^{macro} score, with mean and standard deviation computed across three seeds. While the labeling schemes of *Misviz Synthetic* and *Misviz* largely overlap, specific differences exist. To ensure comparability, the synthetic dataset labels are mapped to the corresponding real-world labels (see Appendix B.3 for details). Accordingly, the out-of-distribution (OOD) rate is reported to account for mismatches where certain classes from *Misviz Synthetic* do not appear in the real-world *Misviz* dataset.

Since the *Misviz* dataset allows for multiple misleaders per chart, but the trained classifiers are designed for multi-class (single-label) prediction, a simplified evaluation strategy is adopted: if the model predicts one of the annotated misleaders, it is considered correct. Otherwise, the misleader with the lowest loss among the ground truth labels is selected as the prediction for that instance. In addition, all chart types not contained in *Misviz Synthetic*, namely *scatter plots*, *maps*, and the *other* category, are removed from the test set. The same is done for not covered misleaders: The misleader *discretized continuous variable* is removed from the test set for evaluation.

4.3.2. Hypotheses

The hypothesis is that models trained on the synthetic training set outperform MLLMs when classifying real-world misleading data visualizations. It is hypothesized that even on the real-world data, the additional axis metadata provided will lead to better performance than the vision encoder-only model. However, it is expected that the models will not achieve comparable results to the evaluation on the test set of the synthetic data, as the real-world data likely includes greater feature variance.

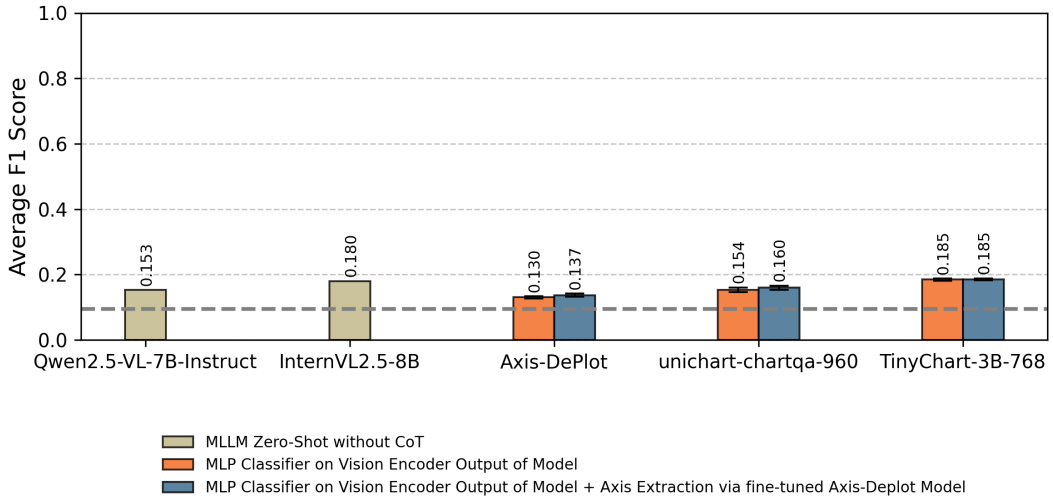


Figure 4.8.: **Macro-Averaged F_1 Score Comparison across Models and Configurations on the *Misviz* Test Set.** The figure shows the mean performance on the test set of the *Misviz* dataset, with macro-averaged F_1 scores computed over three independent runs. The gray dashed line represents the random baseline.

4.3.3. Results

Figure 4.8 presents the macro-averaged F_1 scores for evaluation on the *Misviz* real-world test set. Classifiers trained on the synthetic dataset achieve only modest generalization performance, with most configurations scoring slightly above the random baseline.

Both *TinyChart* encoder based configurations, with and without the axis metadata extracted via *Axis-DePlot* achieve the best results with the average macro-F_1 score being 0.185. The overall best model solely utilizes the vision encoder and achieves a macro-F_1 score of 0.188, beating the model with axis metadata which achieves a score of 0.187. Models incorporating extracted axis metadata outperform their encoder-only counterparts in two out of three configurations, achieving comparable performance in the third. Appendix Table B.3 reports the class-wise F_1 scores for the best-performing model configuration.

Model	Encoder Only	With Axis
UniChart	$4.72\% \pm 0.26\%$	$13.13\% \pm 1.09\%$
TinyChart	$19.02\% \pm 2.89\%$	$20.94\% \pm 0.45\%$
Axis-DePlot	$20.06\% \pm 0.67\%$	$18.51\% \pm 0.26\%$

Table 4.2.: **Mean OOD Rate $\pm$ Standard Deviation across the Three Best-Performing Models (One Selected per Seed) for each Vision Encoder and Model Configuration.** For the MLLMs, no OOD predictions occurred, as the prompts were restricted to misleaders covered by *Misviz*.

The model performance is similar to the MLLM baselines. The *InternVL2.5-8B* model outperforms *Qwen2.5-VL-7B-Instruct* and achieves results comparable to the trained classifiers. Notably, while the best trained classifier slightly exceeds the performance of *InternVL2.5-8B*, it exhibits a high OOD rate of approximately 20%. This

suggests that although synthetic training can match or slightly surpass general-purpose vision-language models on some classes, domain transfer remains a significant challenge.

In addition, Table 4.2 reports the average OOD rate and standard deviation for each vision encoder across both the encoder-only and axis metadata configurations, with values ranging between approximately 4.7% and 20.9% depending on the model.

4.3.4. Error-Analysis

To understand the limited generalization to real-world data, an in-depth error analysis was conducted on the two best-performing trained classifiers on the *Misviz* real-world dataset. The primary aim was to identify and analyze mismatches between the feature distribution of the synthetic dataset and the natural variability present in the real-world dataset. Misclassifications were grouped by their actual and predicted class labels, sorted in descending order of frequency, with potential causes of misclassification manually annotated. In total, 113 images across 27 different mismatch pairs were analyzed. Common sources of error emerged:

Missing Chart Variants

Figure 4.9.: **Examples of Chart Variations Not Covered by *Misviz Synthetic*.** Left: A pie chart containing two misleaders: the slices *misrepresent* the value labels, and the chart is rendered in *3D*. Right: A horizontal bar chart. The synthetic dataset does not include pie charts in *3D* and horizontal bar charts.

A considerable number of misclassifications appear to stem from chart type variations that are not represented in the synthetic dataset. These include horizontal bar charts, stacked bar charts, 3D pie charts, and other complex three-dimensional chart variations. While basic 3D bar charts are included in *Misviz Synthetic*, more diverse 3D bar chart variants, such as those rendered from different viewpoints or with perspective distortions, are not covered. In addition, donut charts, *dual axis* charts with heterogeneous encodings (e.g., bar and line combinations), and non-standard representations (such as human silhouettes or real-world objects) are absent from *Misviz Synthetic*. Among the misclassified examples, 3D pie charts and horizontal bar charts were frequently observed, suggesting a clear distributional gap between the training and real-world datasets (see Figure 4.9). Although 3D bar charts were included during training, they were occasionally misclassified. This may be attributed to the limited diversity of synthetic 3D bar charts, which were rendered from a single viewpoint and thus failed to capture the full range of possible visual perspectives.

Visual and Layout Discrepancies

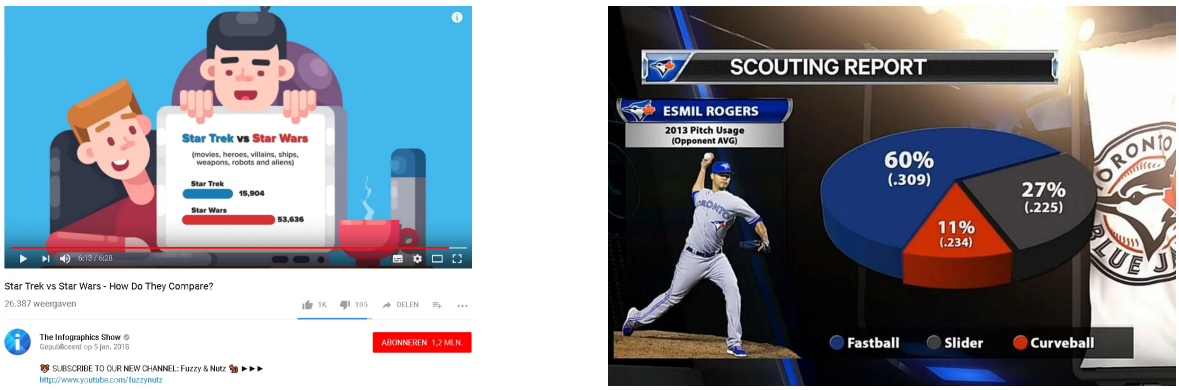


Figure 4.10.: **Examples of Visual Noise and Layout Discrepancies.** Left: A horizontal bar chart exhibiting *misrepresentation*. Right: A *3D* pie chart introducing distortion. In both cases, the chart occupies only part of the image and is surrounded by significant visual noise.

Real-world test images from the *Misviz* dataset often contain complex visual characteristics that are not present in the synthetic images of the *Misviz Synthetic* dataset. These include charts embedded within photographs, screenshots, or physical media, visual clutter, additional descriptive text, and background noise. Furthermore, some charts employ dark color schemes or gradient fills, and many occupy only a portion of the overall image space rather than being centrally positioned (see Figure 4.10). These differences could contribute to misclassifications, particularly if the classifier relies on assumptions such as centralized chart placement or minimal surrounding visual noise. This risk is heightened because all images in the synthetic dataset occupy the whole space in the image and lack visual clutter, which could limit the model's ability to generalize to more complex real-world layouts.

Annotation and Labeling Gaps

When comparing the synthetic dataset to the real-world dataset, the synthetic dataset often lacks fine-grained in-chart annotation diversity, which may contribute to errors in model predictions. In contrast, real-world charts frequently include complex textual and structural elements not represented in the synthetic data. These include non-numeric annotations embedded within the chart, value labels positioned in unconventional locations, categorical labels, and intricate axis formats such as hierarchical (e.g., year + month) or other temporal encodings (e.g. varying date formats). Such variations can complicate the interpretation of chart intent and hinder the accurate identification of misleading design features.

Ambiguous Label Definitions and Missing Labels

While relatively uncommon, some classes exhibited semantic overlap, particularly between *misrepresentation* and *inappropriate use of pie chart*. In a few cases, real-world labels appeared to differ from the annotation conventions used in the synthetic dataset. An additional challenge arose because not all misleading elements present in real-world charts were fully captured by the available labels, and some charts featured ambiguous or subjective distortions that were difficult to classify consistently. These factors may have introduced some degree of noise into the ground truth of the test set, potentially affecting evaluation outcomes.

Axis Metadata Extraction

Since the axis metadata extractor model *Axis-DePlot* is trained exclusively on synthetic data, it inherits the same limitations when applied to real-world data. As a result, it may struggle to accurately extract axis metadata from charts that deviate in structure or appearance from the synthetic training distribution, potentially introducing noise into the classification pipeline (see Figure 4.11).

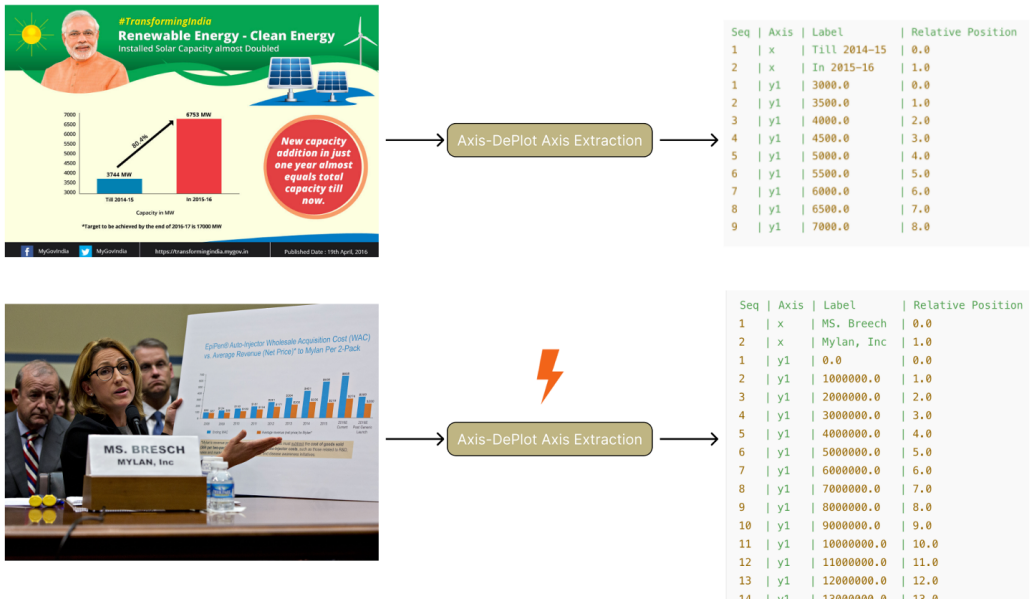


Figure 4.11.: **Examples of Successful and Unsuccessful Axis Metadata Extractions.** The top example shows a successful axis extraction, allowing the classification model to correctly identify the chart as exhibiting a *truncated y-axis*. The bottom example shows a failure in axis metadata extraction: *Axis-DePlot* incorrectly interprets a nameplate within the image as x-axis labels and hallucinates y-axis tick values.

4.3.5. Interim Conclusion

The results from this experiment offer only partial support for the initial hypotheses. While models trained on the *Misviz Synthetic* dataset exhibit some capacity to classify real-world misleading data visualizations, their overall performance remains limited and, in most cases, only slightly above the random baseline. Among all evaluated configurations, only the *TinyChart*-based models achieve a performance level that modestly exceeds that of the strongest MLLM baseline. In contrast, other models perform similarly or fall below it. These findings challenge the assumption that classifiers trained on synthetic data would consistently outperform large-scale pretrained MLLMs in a zero-shot setting.

Incorporating extracted axis metadata via the *Axis-DePlot* model yields modest improvements for two out of three vision encoders, with no observable change for the *TinyChart*-based configurations. This may indicate that axis-related features retain some utility even when applied to real-world data. However, the observed performance gains are modest, indicating that noise introduced by the axis extraction process likely limits the utility of these features.

These results underscore the challenges of transferring models trained on synthetic data to real-world domains. The visual variability and structural complexity of real-world charts, reflected in the accompanying error analysis, contribute to a pronounced feature distribution difference. In response to **RQ3**: *Can models trained on synthetic misleading charts generalize to real-world charts?*, the findings indicate that the models trained on the synthetic data exhibit only limited generalization ability to the real-world data. This highlights the need for improved synthetic data generation processes with broader visual coverage, enhanced modeling strategies, or domain adaptation techniques to better bridge the gap between synthetic and real-world distributions.

5. Conclusion

5.1. Summary of Key Findings

This thesis explored the automatic detection of misleading elements in data visualizations using a vision-based classifier trained on synthetic data. In response to the limitations of rule-based and prompt-driven methods, a new synthetic dataset, *Misviz Synthetic*, was introduced to enable systematic experimentation on misleading data visualizations. The dataset covers 14 types of misleaders across bar, line, and pie charts.

To answer **RQ1**, an ablation study was conducted to evaluate the contribution of different input features. The results show that image representations extracted by vision encoders provide a strong and informative input modality for detecting misleading elements. However, they often lack critical structural information, particularly related to axis properties. Including axis metadata consistently improved classification performance across model configurations, especially for misleaders involving axis manipulation. In contrast, adding the underlying data table provided only marginal gains once axis metadata was already included, suggesting that vision encoders capture much of the tabular context, but underrepresent axis-specific features.

RQ2 was addressed by training classifiers on the synthetic dataset. These models outperformed general-purpose multimodal baselines by a significant margin on the *Misviz Synthetic* test set. The best-performing model combined the *TinyChart* vision encoder with axis metadata extracted via a fine-tuned version of *DePlot* (referred to as *Axis-DePlot*), achieving a macro-averaged F_1 score of 0.782 on the synthetic test set.

To address **RQ3**, the trained models were evaluated on the real-world *Misviz* dataset. While the inclusion of axis metadata yielded slight performance improvements over vision-only models, generalization to real-world data was not achieved. The best-performing configuration reached a macro-averaged F_1 score of 0.188. These results highlight the dataset shift between the synthetic and real-world domain and underscore the difficulty of capturing the visual variability present in real-world charts using synthetic data alone.

5.2. Research Implications

First, the results demonstrate that axis metadata plays a critical role in misleader detection. The performance improvements observed when axis information is included suggest that the classification models benefit from explicit access to structural chart features such as tick values, axis ranges, and scale direction. This indicates a gap in current vision-language pretraining, which may not adequately encode these features.

Second, the use of synthetic data for training and evaluating classifiers demonstrates that synthetic datasets can be valuable when real-world annotations are sparse. *Misviz Synthetic* enabled ablation experiments and model training, offering a scalable approach to benchmark and analyze chart misleader detection. However, the observed performance gap on real-world charts highlights the limitations of relying solely on synthetic

data. This underscores the importance of considering dataset realism and alignment with the real-world data when creating synthetic datasets.

Third, although the classifier does not generalize to real-world charts, its strong performance on `matplotlib`-generated visualizations shows that misleader detection is achievable under structured, synthetic conditions. This confirms that, when misleading elements are consistently defined and rendered, models can learn to identify them with high accuracy. The trained classifier defines a baseline against which future misleader detection models can be compared.

5.3. Limitations

While the results demonstrate the potential of using synthetic data for misleading visualization detection, several limitations should be acknowledged. Although *Misviz Synthetic* covers a range of misleaders, it only captures a fraction of the variability found in real-world visualizations. Key chart types such as horizontal bar charts, donut charts, scatter plots, or 3D pie charts are absent. Furthermore, the dataset is limited to a single misleader per chart, and the classification task is framed as multi-class, reducing expressiveness in cases where multiple misleading elements appear in the same chart. Additionally, the dataset is limited to English-language visualizations, which constrains applicability to global contexts.

No direct comparisons were conducted with chart-specific multimodal models, which may offer stronger classification baselines. While general-purpose SOTA open-source MLLMs in the 7–10B parameter range were evaluated, the absence of chart-domain specific MLLMs leaves open the question of how domain-specific models would perform. The baseline MLLMs classifications approach also did not utilize prompting techniques, which can improve predictive performance, such as CoT approaches. Similarly, the *Axis-DePlot* model was not benchmarked against other MLLMs performing axis metadata extraction through prompting.

The inference speed of *Axis-DePlot* remains relatively slow, which may hinder its applicability in real-time or large-scale settings. Moreover, since the axis metadata extraction model was also trained on synthetic data, it inherits similar limitations observed in the trained classification models. In particular, it struggles with visual styles that are underrepresented in the synthetic training set, such as unconventional fonts, atypical label placements, visual noise, and overlapping chart elements.

5.4. Future Work

Several directions emerge for future research. Expanding *Misviz Synthetic* to include additional chart types,such as horizontal bar charts, 3D pie charts, and donut charts, would help address current gaps in chart type representation. Supporting multiple misleaders per chart and adopting a multi-label classification setup would more accurately reflect the complexity of real-world visualizations. For chart types like pie charts, where real plotting data is difficult to extract, fully synthetic pipelines could be explored to generate misleading data visualizations. Increasing the visual diversity of the dataset by incorporating chart generation libraries beyond `matplotlib` could also help reduce overfitting to a single rendering style and better capture the heterogeneity of real-world charts.

Such improvements to the synthetic dataset would benefit both the classification and axis metadata extraction models, which are trained on the train set of the *Misviz Synthetic*. Furthermore, both components could be

improved by incorporating annotated real-world visualizations to strengthen model generalization beyond synthetic data.

Future work could explore few-shot axis metadata extraction with MLLMs, as it could potentially offer a more flexible and better-performing alternative to the proposed axis extraction approach. The classifier performance may also benefit from incorporating vision model architectures not evaluated in this work, such as vision tower encoders or multi-encoder vision models. These vision encoder approaches may better capture structural chart elements, particularly axis metadata, within their visual representations.

Integrating preprocessing steps could further enhance classification robustness. Chart detection may help isolate relevant regions in noisy or cluttered images, while chart-type classification could constrain the set of potential misleading factors, improving both efficiency and accuracy in downstream detection.

Given its observed impact on detection performance, future work should explore vision encoder pretraining tasks that more explicitly model axis-related features such as scales, tick positions, and label orientation. Encouraging models to encode these structural elements may improve their ability to detect misleading data visualizations. Developing such objectives may help models learn structural chart features more effectively, supporting improved generalization and performance on downstream chart understanding tasks.

Building on this foundation, future work may focus on developing systems that go beyond detection to also explain and correct misleading visualizations, such as by restoring truncated axes or removing 3D effects. This research represents a foundational step toward intelligent tools that promote critical engagement with data visualizations and empower users to defend against manipulation.

Bibliography

[1] Zeit Online. *Bundestagswahl 2021: Ergebnisse nach Gemeinden*. 2021. URL: https://www.zeit.de/politik/deutschland/2021-09/ergebnisse-bundestagswahl-gemeinde-karte (Accessed on 15.4.2025).

[2] Tagesschau. *Ergebnisse der Bundestagswahl 2021*. 2025. URL: https://www.tagesschau.de/inland/bundestagswahl/ergebnisse (Accessed on 15.4.2025).

[3] The New York Times. *Graphics and Multimedia - The New York Times*. URL: https://www.nytimes.com/spotlight/graphics (Accessed on 15.4.2025).

[4] Visual Capitalist. *Visualizing the World's Data*. URL: https://www.visualcapitalist.com/ (Accessed on 15.4.2025).

[5] Leo Yu-Ho Lo, Ayush Gupta, Kento Shigyo, Aoyu Wu, Enrico Bertini, and Huamin Qu. "Misinformed by Visualization: What Do We Learn From Misinformative Visualizations?" In: *Computer Graphics Forum* 41.3 (2022), pp. 515–525. DOI: 10.1111/cgf.14559.

[6] Maxim Lisnic, Cole Polychronis, Alexander Lex, and Marina Kogan. "Misleading Beyond Visual Tricks: How People Actually Lie with Charts". In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 2023, pp. 1–21. DOI: 10.1145/3544548.3580910.

[7] Lily W. Ge, Yuan Cui, and Matthew Kay. "CALVI: Critical Thinking Assessment for Literacy in Visualizations". In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 2023, pp. 1–18. DOI: 10.1145/3544548.3581406.

[8] Lianna Brinded. *I'm starting to think a 'Brexit' is a good idea and I never thought I'd ever say that*. 2015. URL: https://www.businessinsider.com/reasons-why-uk-leaving-the-eu-brexit-is-a-good-idea-2015-10?IR=T&r=UK (Accessed on 15.4.2025).

[9] White House. *America's Economic Growth in the 21st Century*. Tweet from the Official White House X Account @WhiteHouse46 (Archived). 2022. URL: https://x.com/WhiteHouse46/status/1486709480351952901 (Accessed on 15.4.2025).

[10] Marc Lallanilla. *Misleading Gun-Death Chart Draws Fire*. 2014. URL: https://www.livescience.com/45083-misleading-gun-death-chart.html (Accessed on 15.4.2025).

[11] Mubashara Akhtar, Nikesh Subedi, Vivek Gupta, Sahar Tahmasebi, Oana Cocarascu, and Elena Simperl. "ChartCheck: Explainable Fact-Checking over Real-World Chart Images". In: *Findings of the Association for Computational Linguistics: ACL 2024*. 2024, pp. 13921–13937. DOI: 10.18653/v1/2024.findings-acl.828.

[12] OpenAI et al. *GPT-4 Technical Report*. 2024. DOI: 10.48550/arXiv.2303.08774.

[13] Qwen Team. *Qwen2.5-VL*. 2025. URL: https://qwenlm.github.io/blog/qwen2.5-vl/.

[14] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. “Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 24185–24198.

[15] Andrew McNutt and Gordon Kindlmann. “Linting for Visualization: Towards a Practical Automated Visualization Guidance System”. In: *VisGuides: 2nd Workshop on the Creation, Curation, Critique*. 2018.

[16] Andrew McNutt, Gordon Kindlmann, and Michael Correll. “Surfacing Visualization Mirages”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–16. DOI: 10.1145/3313831.3376420.

[17] Shubham Bharti, Shiyun Cheng, Jihyun Rho, Martina Rao, and Xiaojin Zhu. “CHARTOM: A Visual Theory-of-Mind Benchmark for Multimodal Large Language Models”. In: *arXiv preprint arXiv:2408.14419* (2024). DOI: 10.48550/arXiv.2408.14419.

[18] Jihyun Rho, Martina Rau, Shubham Kumar Bharti, Rosanne Luu, Jeremy McMahan, Andrew Wang, and Xiaojin Zhu. “Various Misleading Visual Features in Misleading Graphs: Do they truly deceive us?” In: *Proceedings of the Annual Meeting of the Cognitive Science Society 46*. 2024. URL: https://escholarship.org/uc/item/0kk6b4cn (Accessed on 3.5.2025).

[19] Jonathan Tonglet, Tinne Tuytelaars, Marie-Francine Moens, and Iryna Gurevych. “Protecting multimodal LLMs against misleading visualizations”. In: *arXiv preprint arXiv:2502.20503* (2025). DOI: 10.48550/arXiv.2502.20503.

[20] Saugat Pandey and Alvitta Ottley. “Benchmarking Visual Language Models on Standardized Visualization Literacy Tests”. In: *Computer Graphics Forum* 44.3 (2025). DOI: 10.1111/cgf.15000.

[21] Jason Alexander, Priyal Nanda, Kai-Cheng Yang, and Ali Sarvghad. “Can GPT-4 Models Detect Misleading Visualizations?” In: *2024 IEEE Visualization and Visual Analytics (VIS)*. 2024, pp. 106–110. DOI: 10.1109/VIS55277.2024.00029.

[22] Leo Yu-Ho Lo and Huamin Qu. “How Good (Or Bad) Are LLMs at Detecting Misleading Visualizations?” In: *IEEE Transactions on Visualization and Computer Graphics* (2024), pp. 1–10. DOI: 10.1109/TVCG.2024.3456333.

[23] World Economic Forum. *Global Risks Report 2025: Conflict, Environment and Disinformation Top Threats*. 2025. URL: https://www.weforum.org/press/2025/01/global-risks-report-2025-conflict-environment-and-disinformation-top-threats/ (Accessed on 7.4.2025).

[24] Vahed Qazvinian, Emily Rosengren, Dragomir R. Radev, and Qiaozhu Mei. “Rumor has it: identifying misinformation in microblogs”. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2011, pp. 1589–1599. DOI: 10.5555/2145432.2145602.

[25] Claire Wardle and Hossein Derakhshan. *Information disorder: Toward an interdisciplinary framework for research and policy making*. Tech. rep. DGI (2017) 09. Council of Europe, 2017. URL: https://rm.coe.int/information-disorder-toward-an-interdisciplinary-framework-for-researc/168076277c.

[26] Yangming Zhou, Yuzhou Yang, Qichao Ying, Zhenxing Qian, and Xinpeng Zhang. “Multimodal Fake News Detection via CLIP-Guided Learning”. In: *2023 IEEE International Conference on Multimedia and Expo (ICME)*. 2023, pp. 2825–2830. DOI: 10.1109/ICME55011.2023.00480.

[27] Kai Nakamura, Sharon Levy, and William Yang Wang. “Fakeddit: A New Multimodal Benchmark Dataset for Fine-grained Fake News Detection”. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. 2020, pp. 6149–6157. DOI: 10.48550/arXiv.1911.03854.

[28] Yixuan Li, Xuelin Liu, Xiaoyang Wang, Shiqi Wang, and Weisi Lin. “FakeBench: Uncover the Achilles’ Heels of Fake Images with Large Multimodal Models”. In: *arXiv preprint arXiv:2404.13306* (2024). DOI: 10.48550/arXiv.2404.13306.

[29] Runsheng Huang, Liam Dugan, Yue Yang, and Chris Callison-Burch. “MiRAGeNews: Multimodal Realistic AI-Generated News Detection”. In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. 2024, pp. 16436–16448. DOI: 10.18653/v1/2024.findings-emnlp.959.

[30] Roberto Amoroso, Davide Morelli, Marcella Cornia, Lorenzo Baraldi, Alberto Del Bimbo, and Rita Cucchiara. “Parents and Children: Distinguishing Multimodal Deepfakes from Natural Images”. In: *ACM Trans. Multimedia Comput. Commun. Appl.* 21.1 (2024). DOI: 10.1145/3665497.

[31] Anshul Vikram Pandey, Katharina Rall, Margaret L. Satterthwaite, Oded Nov, and Enrico 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*. 2015, pp. 1469–1478. DOI: 10.1145/2702123.2702608.

[32] Arlen Fan, Yuxin Ma, Michelle Mancenido, and Ross Maciejewski. “Annotating Line Charts for Addressing Deception”. In: *CHI Conference on Human Factors in Computing Systems*. 2022, pp. 1–12. DOI: 10.1145/3491102.3502138.

[33] Xingyu Lan and Yu Liu. ““I Came Across a Junk”: Understanding Design Flaws of Data Visualization from the Public’s Perspective”. In: *IEEE Transactions on Visualization and Computer Graphics* 31.1 (2025), pp. 393–403. DOI: 10.1109/TVCG.2024.3456341.

[34] Anshul Vikram Pandey, Anjali Manivannan, Oded Nov, Margaret Satterthwaite, and Enrico Bertini. “The Persuasive Power of Data Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2211–2220. DOI: 10.1109/TVCG.2014.2346419.

[35] Shaun O’Brien and Claire Lauer. “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*. ACM, 2018, pp. 1–8. DOI: 10.1145/3233756.3233961.

[36] Sukwon Lee, Sung-Hee Kim, and Bum Chul Kwon. “VLAT: Development of a Visualization Literacy Assessment Test”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 551–560. DOI: 10.1109/TVCG.2016.2598920.

[37] Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Akos Kadar, Adam Trischler, and Yoshua Bengio. *FigureQA: An Annotated Figure Dataset for Visual Reasoning*. Workshop paper at International Conference on Learning Representations (ICLR) 2018. 2018. DOI: 10.48550/arXiv.1710.07300.

[38] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. “DVQA: Understanding Data Visualizations via Question Answering”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5648–5656. DOI: 10.48550/arXiv.1801.08163.

[39] Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. “PlotQA: Reasoning over Scientific Plots”. In: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2020, pp. 1516–1525. DOI: 10.1109/WACV45572.2020.9093523.

[40] Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. “ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning”. In: *Findings of the Association for Computational Linguistics: ACL 2022*. 2022, pp. 2263–2279. DOI: 10.18653/v1/2022.findings-acl.177.

[41] Shankar Kantharaj, Xuan Long Do, Rixie Tiffany Leong, Jia Qing Tan, Enamul Hoque, and Shafiq Joty. "OpenCQA: Open-ended Question Answering with Charts". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 2022, pp. 11817–11837. DOI: 10.18653/v1/2022.emnlp-main.811.

[42] Fangyu Liu, Julian Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. "DePlot: One-shot visual language reasoning by plot-to-table translation". In: *Findings of the Association for Computational Linguistics: ACL 2023*. 2023, pp. 10381–10399. DOI: 10.18653/v1/2023.findings-acl.660.

[43] Junyu Luo, Zekun Li, Jinpeng Wang, and Chin-Yew Lin. "ChartOCR: Data Extraction from Charts Images via a Deep Hybrid Framework". In: *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2021, pp. 1916–1924. DOI: 10.1109/WACV48630.2021.00196.

[44] Ting-Yao Hsu, C Lee Giles, and Ting-Hao Huang. "SciCap: Generating Captions for Scientific Figures". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. 2021, pp. 3258–3264. DOI: 10.18653/v1/2021.findings-emnlp.277.

[45] E. Hoque, P. Kavehzadeh, and A. Masry. "Chart Question Answering: State of the Art and Future Directions". In: *Computer Graphics Forum* 41.3 (2022), pp. 555–572. DOI: https://doi.org/10.1111/cgf.14573.

[46] Shankar Kantharaj, Rixie Tiffany Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. "Chart-to-Text: A Large-Scale Benchmark for Chart Summarization". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2022, pp. 4005–4023. DOI: 10.18653/v1/2022.acl-long.277.

[47] Benny Tang, Angie Boggust, and Arvind Satyanarayan. "VisText: A Benchmark for Semantically Rich Chart Captioning". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2023, pp. 7268–7298. DOI: 10.18653/v1/2023.acl-long.401.

[48] Kung-Hsiang Huang, Hou Pong Chan, May Fung, Haoyi Qiu, Mingyang Zhou, Shafiq Joty, Shih-Fu Chang, and Heng Ji. "From Pixels to Insights: A Survey on Automatic Chart Understanding in the Era of Large Foundation Models". In: *IEEE Transactions on Knowledge and Data Engineering* 37.5 (2025), pp. 2550–2568. DOI: 10.1109/TKDE.2024.3513320.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017. DOI: 10.5555/3295222.3295349.

[50] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[51] Jeffrey L. Elman. "Finding structure in time". In: *Cognitive Science* 14.2 (1990), pp. 179–211. DOI: https://doi.org/10.1016/0364-0213(90)90002-E.

[52] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[53] Alexander Kolesnikov, Alexey Dosovitskiy, Dirk Weissenborn, Georg Heigold, Jakob Uszkoreit, Lucas Beyer, Matthias Minderer, Mostafa Dehghani, Neil Houlsby, Sylvain Gelly, Thomas Unterthiner, and Xiaohua Zhai. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *Proceedings of the 9th International Conference on Learning Representations (ICLR)*. 2021. DOI: 10.48550/arXiv.2010.11929.

[54] Ibrahim Alabdulmohsin, Xiaohua Zhai, Alexander Kolesnikov, and Lucas Beyer. “Getting ViT in shape: scaling laws for compute-optimal model design”. In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. 2023. DOI: 10.5555/3666122.3666844.

[55] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows”. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 9992–10002. DOI: 10.1109/ICCV48922.2021.00986.

[56] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. “Towards VQA Models That Can Read”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 8309–8318. DOI: 10.1109/CVPR.2019.00851.

[57] Pan Zhang et al. *InternLM-XComposer: A Vision-Language Large Model for Advanced Text-image Comprehension and Composition*. 2023. DOI: 10.48550/arXiv.2309.15112.

[58] Anwen Hu, Shizhe Chen, and Qin Jin. “Question-controlled Text-aware Image Captioning”. In: *Proceedings of the 29th ACM International Conference on Multimedia*. 2021, pp. 3097–3105. DOI: 10.1145/3474085.3475452.

[59] Jiayun Fu, Bin B. Zhu, Haidong Zhang, Yayi Zou, Song Ge, Weiwei Cui, Yun Wang, Dongmei Zhang, Xiaojing Ma, and Hai Jin. “ChartStamp: Robust Chart Embedding for Real-World Applications”. In: *Proceedings of the 30th ACM International Conference on Multimedia*. 2022, pp. 2786–2795. DOI: 10.1145/3503161.3548286.

[60] Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. “ChartLlama: A Multimodal LLM for Chart Understanding and Generation”. In: *arXiv preprint arXiv:2311.16483* (2023). DOI: 10.48550/arXiv.2311.16483.

[61] Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Eisenschlos. “MatCha: Enhancing Visual Language Pretraining with Math Reasoning and Chart Derendering”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2023, pp. 12756–12770. DOI: 10.18653/v1/2023.acl-long.714.

[62] Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. “UniChart: A Universal Vision-language Pretrained Model for Chart Comprehension and Reasoning”. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 2023, pp. 14662–14684. DOI: 10.18653/v1/2023.emnlp-main.906.

[63] Ahmed Masry, Mehrad Shahmohammadi, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. “ChartInstruct: Instruction Tuning for Chart Comprehension and Reasoning”. In: *Findings of the Association for Computational Linguistics: ACL 2024*. 2024, pp. 10387–10409. DOI: 10.18653/v1/2024.findings-acl.619.

[64] Fanqing Meng, Wenqi Shao, Quanfeng Lu, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. “ChartAssistant: A Universal Chart Multimodal Language Model via Chart-to-Table Pre-training and Multitask Instruction Tuning”. In: *Findings of the Association for Computational Linguistics: ACL 2024*. 2024, pp. 7775–7803. DOI: 10.18653/v1/2024.findings-acl.463.

[65] Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. “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*. 2024, pp. 1882–1898. DOI: 10.18653/v1/2024.emnlp-main.112.

[66] Fuxiao Liu, Xiaoyang Wang, Wenlin Yao, Jianshu Chen, Kaiqiang Song, Sangwoo Cho, Yaser Yacoob, and Dong Yu. “MMC: Advancing Multimodal Chart Understanding with Large-scale Instruction Tuning”. In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 2024, pp. 1287–1310. DOI: 10.18653/v1/2024.naacl-long.70.

[67] Renqiu Xia, Haoyang Peng, Hancheng Ye, Mingsheng Li, Xiangchao Yan, Peng Ye, Botian Shi, Yu Qiao, Junchi Yan, and Bo Zhang. “StructChart: On the Schema, Metric, and Augmentation for Visual Chart Understanding”. In: *arXiv preprint arXiv:2309.11268* (2024). DOI: 10.48550/arXiv.2309.11268.

[68] OpenAI. *GPT-4o Mini: Advancing Cost-Efficient Intelligence*. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence. 2024. (Accessed on 14.4.2025).

[69] OpenAI. *Hello GPT-4o*. https://openai.com/index/hello-gpt-4o. 2024. (Accessed on 14.4.2025).

[70] OpenAI. *GPT-4V(ision) System Card*. https://openai.com/research/gpt-4v-system-card. 2023. (Accessed on 14.4.2025).

[71] Evanthia Dimara, Steven Franconeri, Catherine Plaisant, Anastasia Bezerianos, and Pierre Dragicevic. “A Task-Based Taxonomy of Cognitive Biases for Information Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 26.2 (2020), pp. 1413–1432. DOI: 10.1109/TVCG.2018.2872577.

[72] Max Roser, Esteban Ortiz-Ospina, and Hannah Ritchie. *Our World in Data*. https://ourworldindata.org. 2025. (Accessed on 15.4.2025).

[73] Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. “TabFact: A Large-scale Dataset for Table-based Fact Verification”. In: *Proceedings of the 8th International Conference on Learning Representations (ICLR)*. 2020. DOI: 10.48550/arXiv.1909.02164.

[74] Creative Commons. *Attribution 4.0 International (CC BY 4.0) License*. 2013. URL: https://creativecommons.org/licenses/by/4.0/ (Accessed on 1.5.2025).

[75] John D. Hunter. “Matplotlib: A 2D Graphics Environment”. In: *Computing in Science Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[76] Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. “TaPas: Weakly Supervised Table Parsing via Pre-training”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 4320–4333. DOI: 10.18653/v1/2020.acl-main.398.

[77] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. “Learning Transferable Visual Models From Natural Language Supervision”. In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. 2021, pp. 8748–8763. DOI: 10.48550/arXiv.2103.00020.

[78] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. “Sigmoid Loss for Language Image Pre-Training”. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 11941–11952. DOI: 10.1109/ICCV51070.2023.01100.

[79] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. “Token Merging: Your ViT But Faster”. In: *Proceedings of the 11th International Conference on Learning Representations (ICLR)*. 2023. DOI: 10.48550/arXiv.2210.09461.

[80] Ahmed Masry, Megh Thakkar, Aayush Bajaj, Aaryaman Kartha, Enamul Hoque, and Shafiq Joty. "ChartGemma: Visual Instruction-tuning for Chart Reasoning in the Wild". In: *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*. 2025, pp. 625–643. DOI: 10.48550/arXiv.2407.04172.

[81] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. "Improved Baselines with Visual Instruction Tuning". In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 26286–26296. DOI: 10.1109/CVPR52733.2024.02484.

[82] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. 2015. DOI: 10.48550/arXiv.1412.6980.

[83] Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. "Pix2Struct: screenshot parsing as pretraining for visual language understanding". In: *Proceedings of the 40th International Conference on Machine Learning*. 2023. DOI: 10.48550/arXiv.2210.03347.

[84] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. "LoRA: Low-Rank Adaptation of Large Language Models". In: *International Conference on Learning Representations (ICLR)*. 2022. DOI: 10.48550/arXiv.2106.09685.

[85] Noam Shazeer and Mitchell Stern. "Adafactor: Adaptive Learning Rates with Sublinear Memory Cost". In: *Proceedings of the 35th International Conference on Machine Learning*. 2018. DOI: 10.48550/arXiv.1804.04235.

[86] Anthropic. *Claude 3.7 Sonnet*. https://claude.ai. 2025. (Accessed on 1.5.2025).

A. Additional Information Synthetic Dataset

A.1. Misleader Overview

Table A.1.: The misleaders included in the *Misviz Synthetic* dataset and their definition. Arrows indicate, that the misleader is seen as a subtype of the previously listed misleader without an arrow. Arrows in brackets indicate that only under special conditions the misleader can be seen as a subtype (see special remarks)

Misleader	**Definition**	**Special Cases and Remarks**
No Misleader	A chart without any misleaders.	
Inappropriate Item Order	Items are arranged in an unconventional order, misleading the audience or creating confusion (Adapted from [5]).	
→ Inverted X-Axis	The x-axis is oriented in an unconventional direction, and the perception of the data is reversed (Adapted from [5]).	Can also be classified as "Inappropriate item order".
Misrepresentation	Misrepresentation occurs when the value labels provided do not match the visual encoding [5].	
(→) Nonlinear Y-Axis	Scaling of the y-axis of the data changes in the middle of the axis.	In case that the chart is plotted without a y-axis, it depicts a misrepresentation.
(→) Truncated Y-Axis	The axis does not start from zero or is truncated in the middle, resulting in an exaggerated difference between bars of different values (Adapted from [5]).	In case that the chart is plotted without a y-axis and only two records are present, it depicts a misrepresentation.
Inappropriate Use of Pie Chart (Not Sum of One)	When a pie chart is used for non-part-to-whole data (Adapted from [5]).	
Inverted Y-Axis	The y-axis is oriented in an unconventional direction and the perception of the data is reversed (Adapted from [5]).	

Misleader	Definition	Special Cases and Remarks
Inconsistent Binning Size	Inconsistent binning size occurs when there are variations in the boundaries of the binning groups of a chart with data bins (Adapted from [5]).	
Inappropriate Use of Accumulation	A cumulative measure is used on data to hide a declining trend in the data (Adapted from [5]).	
Inconsistent intervals	Inconsistent axis ticks refer to cases with varying intervals between the ticks [5].	
3D	A chart is plotted in 3D, making data representations appear distorted. For 3D, the closer something is, the larger it appears, despite being the same size in 3D perspective (Adapted from [5]).	
Inappropriate Axis Range	In the case of an inappropriate axis range, the axis range is either too broad or too narrow to accurately visualize the data, allowing changes to be minimized or maximized depending on the author's intention [5].	Exclusively applied to single axis charts.
Dual Axis	Dual axis is when two independent axes are layered on top of each other with inappropriate scaling. This results in a misleading narrative about the relationship between the two. However, other misleaders such as inappropriate axis range might apply as well [5].	Other misleaders might apply as well.
Inappropriate use of line chart	A line chart is deemed inappropriate when used in an unconventional way or in a way that results in incorrect interpretation of the data or intentionally misleading the audience [5].	

A.2. Misleader Plotting Pipeline Visualization

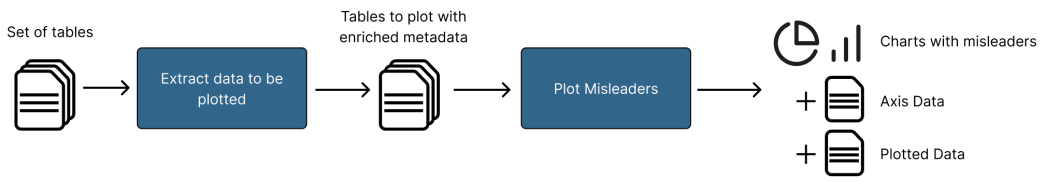


Figure A.1.: General overview of the plotting steps of the synthetic data generation pipeline with the associated output of each step.

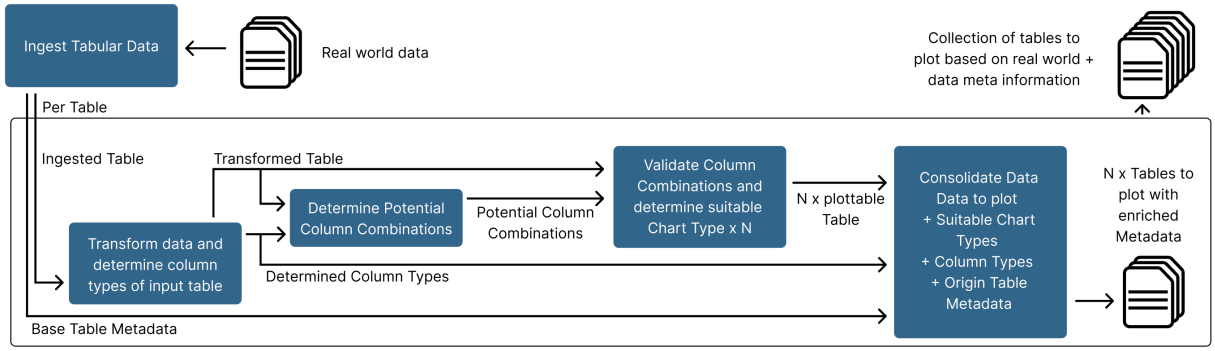


Figure A.2.: Overview of the first step of the plotting pipeline, which extracts data to be plotted from the data input. Appropriate columns are combined, chart titles are determined, and appropriate chart types are determined.

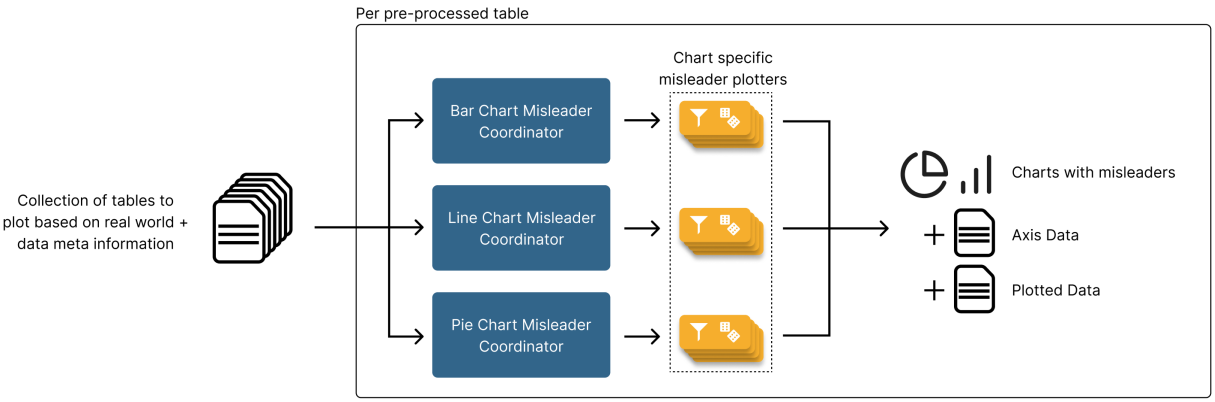


Figure A.3.: Overview of the second stage of the plotting pipeline, which generates misleading data visualizations. The input data is routed to chart-specific *misleader plotters* that apply appropriate misleaders based on data structure and content. Random variations are applied to each chart to introduce visual diversity. For all generated charts, the manipulated underlying data is saved. In addition, axis metadata is extracted for coordinate-based chart types to support downstream processing.

A.3. Misleader Example Images and Implementation

This section defines each misleader category in the dataset and describes the specific implementation strategies used to simulate each misleading pattern in the synthetic data.

A.3.1. No Misleader

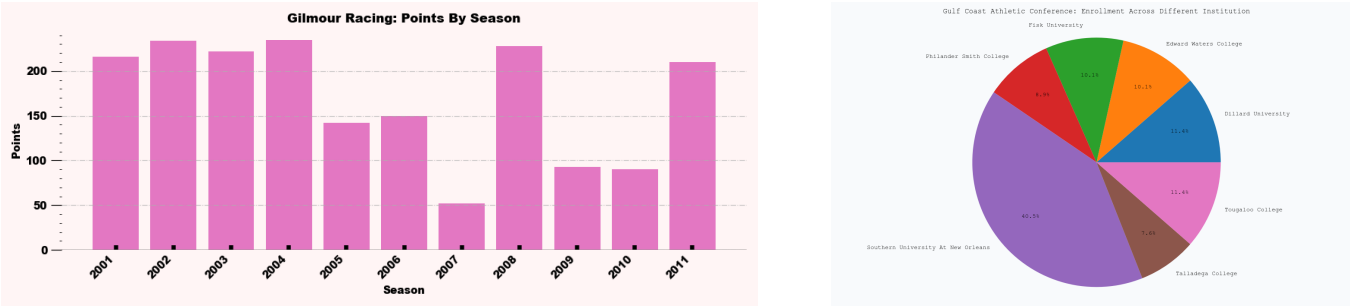


Figure A.4.: Example of a bar chart and a pie chart without misleading elements.

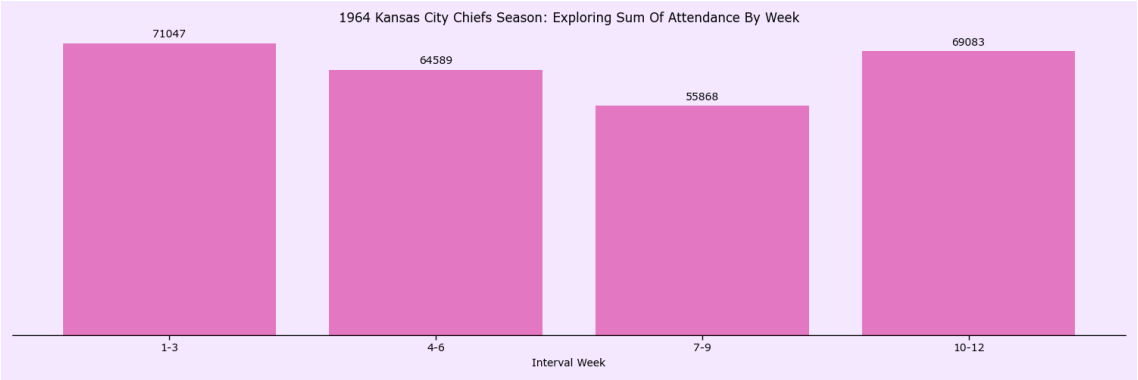


Figure A.5.: Example of a binned chart without misleading elements.

Definition
A chart without any misleaders.

Implementation
Non-misleading charts are plotted for all chart types by ensuring that no misleading factors are present in the underlying data. For example, the step size between time values must remain consistent when using a temporal independent variable to avoid introducing implicit distortions.

In addition, for specific misleading factors such as *inconsistent binning size*, non-misleading counterparts are explicitly required to prevent models from learning to associate binned charts with misleading behavior by default. Non-misleading charts are implemented for all chart types. For bar charts, both non-binned and correctly binned bar charts are generated to ensure proper coverage.

A.3.2. Inappropriate Item Order

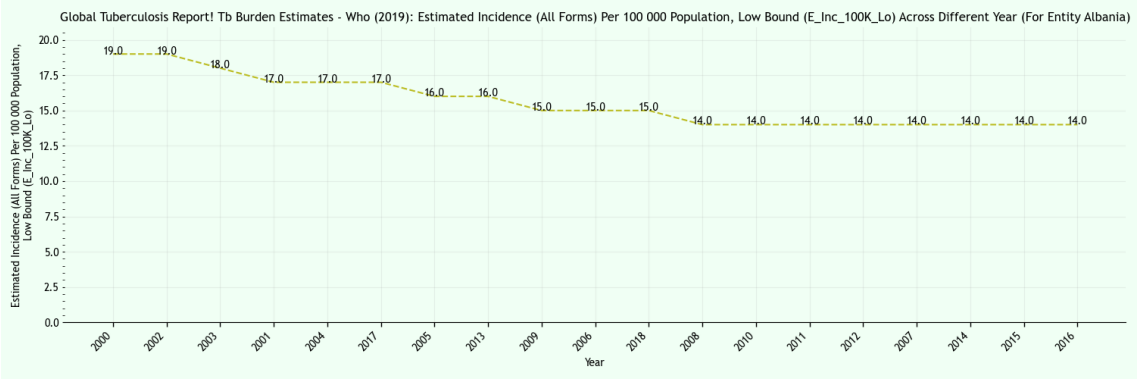


Figure A.6.: An example for *inappropriate item ordering* in a line chart.

Definition
Items are arranged in an unconventional order, misleading the audience or creating confusion (adapted from [5]).

Implementation
This misleader is applied to charts with temporal values on the x-axis. The implementation checks whether the temporal values are sorted in increasing or decreasing order. If they are not, the chart is randomly sorted in ascending or descending order based on the associated values before plotting, resulting in a non-chronological x-axis and a misleading impression of the data.

A.3.3. Inverted X-Axis

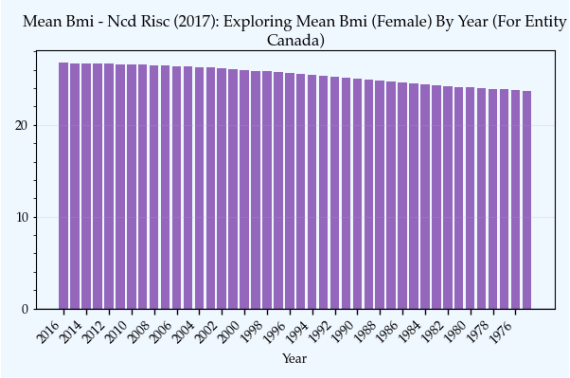


Figure A.7.: Example for a data visualizations applying the misleader *inverted x-axis*.

Definition
The x-axis is oriented in an unconventional direction and the perception of the data is reversed. Can also be classified as *inappropriate item order* (adapted from [5]).

Implementation

Given a temporal x-axis, the tick labels are sorted from left to right in descending order of the independent variable.

A.3.4. Misrepresentation

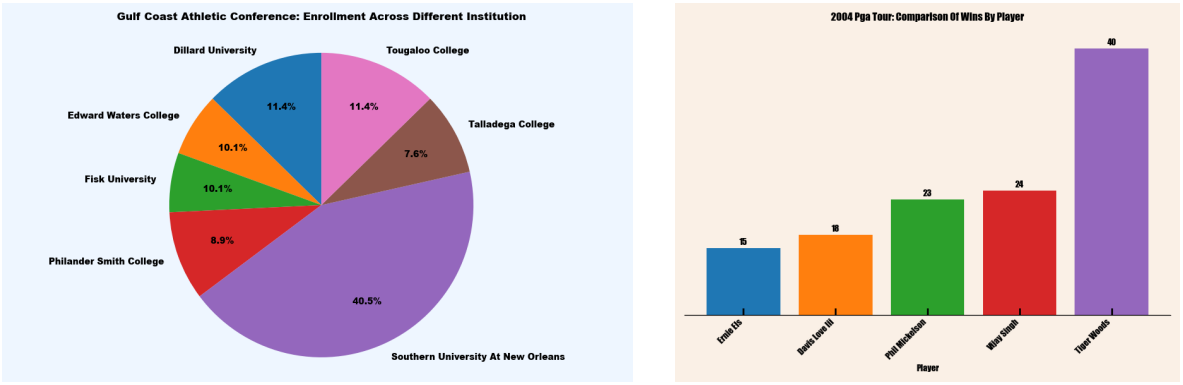


Figure A.8.: Examples for data visualizations with the *misrepresentation* misleader applied.

Definition
Misrepresentation occurs when the value labels provided do not match the visual encoding [5].

Implementation
The misleader is applied across all chart types. First, the input data is preserved. Then, each dependent numerical variable is randomly scaled by one of the following factors: [0.65, 0.7, 0.75, 0.8, 1.2, 1.25, 1.3, 1.35]. The chart is plotted using the manipulated values, while the original (unscaled) values are displayed as value labels. This misleader depends on the presence of value labels, as the misleading effect arises from the discrepancy between the visual encoding and the shown numerical values.

A.3.5. Nonlinear Y-Axis

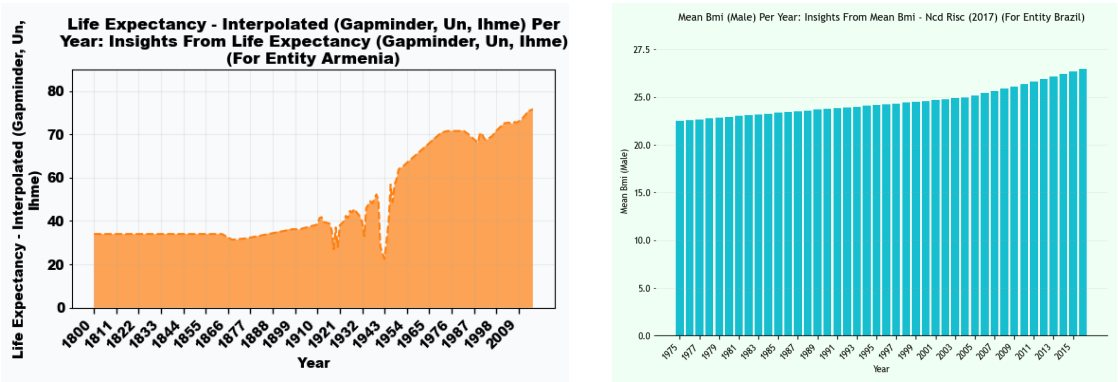


Figure A.9.: Examples for data visualizations with the misleader *nonlinear y-axis* applied.

Definition
Scaling of the y-axis of the data changes in the middle of the axis. In case that the chart is plotted without a y-axis, it depicts a misrepresentation.

Implementation
This misleader is applied to coordinate-based data visualizations. Charts where the data values are too similar are excluded (specifically, the minimum to maximum value ratio must be below 90%), as the manipulation would not produce a strong visual effect. The value with the most significant jump to the next highest value is identified to determine where the scale should change, and the original tick location closest to this point is used as the scale transition. After this point, the y-axis step size is reduced by half, creating a nonlinear scale that visually distorts the relative magnitude of the higher values.

A.3.6. Truncated Y-Axis

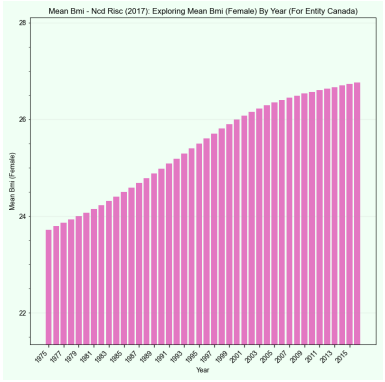


Figure A.10.: Example for a data visualization with the misleader *truncated y-axis* applied.

Definition
The axis does not start from zero or is truncated in the middle, resulting in an exaggerated difference between bars of different values (adapted from [5]). In case that the chart is plotted without a y-axis and only two records are present, it depicts a misrepresentation.

Implementation
This misleader is applied to bar charts. Input tables are filtered to retain only positive values. To ensure that the truncated y-axis produces a noticeable visual effect, the ratio of minimum to maximum value is computed, and the smallest value must be at least 50% of the largest value for the chart to qualify. This constraint ensures that the truncation exaggerates the perceived difference between otherwise similar values.

A.3.7. Inappropriate Use of Pie Chart

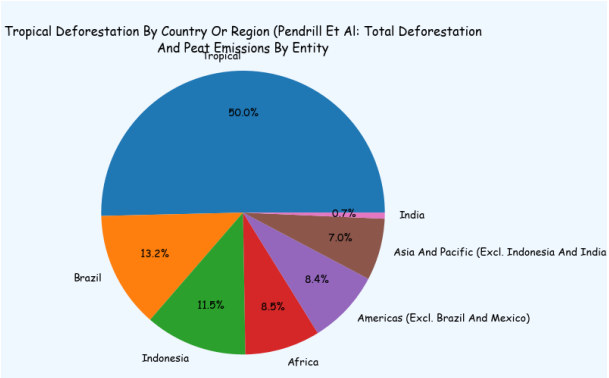


Figure A.11.: Example for a data visualization where a pie chart is used inappropriately. The percentages do not sum to 100%.

Definition
A pie chart contains the misleader *inappropriate use of pie chart* when a pie chart is used for non-part-to-whole data (adapted from [5]).

Implementation
This misleader covers two scenarios of misrepresentation. In the first case, the relative shares of each value are calculated based on the complete input data. One data record is removed, and the chart is plotted using the reduced dataset. However, the original share values, calculated before removal, are retained as value labels. In the second case, if an input table contains positive percentage-based values that do not sum to 1 (or 100%), a pie chart is plotted using the literal percentage values as labels.

A.3.8. Inverted Y-Axis

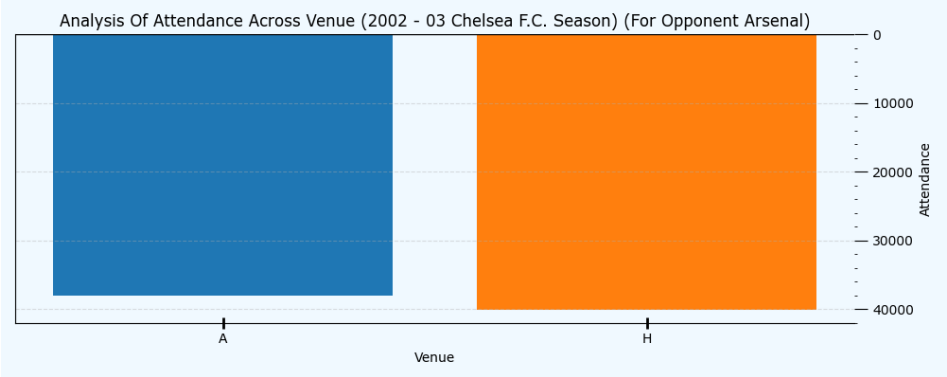


Figure A.12.: Example for a data visualization with an *inverted y-axis*.

Definition
The y-axis is oriented in an unconventional direction and the perception of the data is reversed (adapted from [5]).

Implementation
This misleader is applied only to coordinate-based charts by inverting the y-axis.

A.3.9. Inconsistent Binning Size

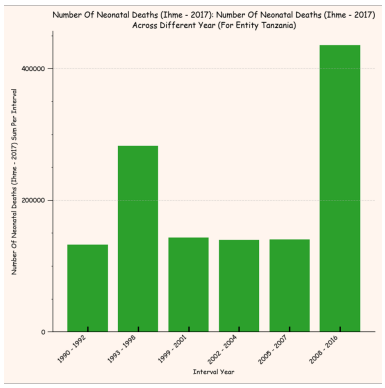


Figure A.13.: An example for *inconsistent binning size*.

Definition
Inconsistent binning size occurs when there are variations in the boundaries of the binning groups of a chart with data bins (adapted from [5]).

Implementation
This misleader is only applied to bar charts with temporal values on the x-axis. It differentiates between sum and average aggregation when binning data over time intervals. If the values to be plotted contain percentage-like entries (i.e., the values sum to one or one hundred), the data is averaged. Otherwise, the values are summed. The formatting of the temporal intervals is randomized based on predefined templates. Additionally, the chart title is updated to explicitly include either *sum* or *average*, depending on the applied aggregation method.

A.3.10. Inappropriate Use of Accumulation

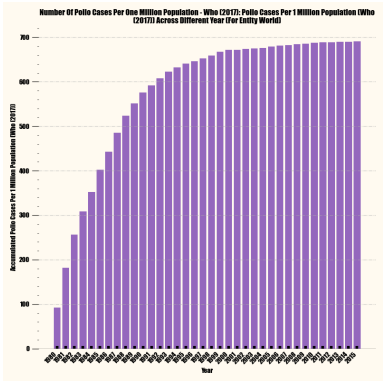


Figure A.14.: An example for *inappropriate use of accumulation*.

Definition
A cumulative measure is used on data to hide a declining trend in the data (adapted from [5]).

Implementation
This misleader is applied to bar charts with temporal values on the x-axis. If a downward trend is detected in the data, the values are transformed into a cumulative sum. The y-axis label is updated accordingly by prefixing the original column name with *accumulated* to reflect the manipulation.

A.3.11. Inconsistent Intervals

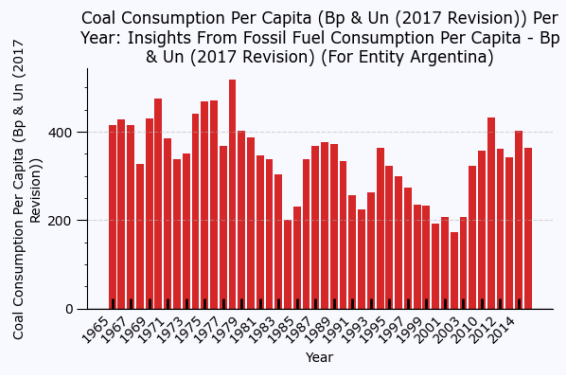


Figure A.15.: An example for a data visualization containing the misleader *inconsistent axis intervals.*

Definition
Inconsistent axis ticks refer to cases with varying intervals between the ticks [5].

Implementation
This misleader is applied only to coordinate-based chart types with temporal values on the x-axis. It is implemented by removing a random number of records from the input table, resulting in irregular gaps between x-axis ticks. For tables with seven or more records, up to 30% of the data, limited to a maximum of five consecutive records near the center of the temporal sequence, is removed. This creates visually inconsistent intervals along the x-axis in the resulting chart.

A.3.12. 3D

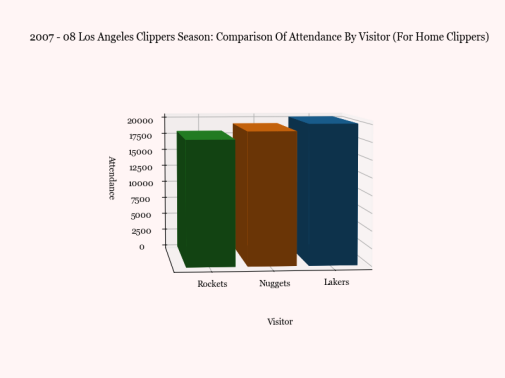


Figure A.16.: Example for a data visualization with a 3D-effect applied.

Definition
A chart is plotted in 3D, making data representations appear distorted. For 3D, the closer something is, the larger it appears, despite being the same size in 3D perspective (adapted from [5]).

Implementation
This misleader is implemented only for bar charts, as `matplotlib` does not provide straightforward support for rendering 3D pie or line charts. To avoid label overlap caused by rotation, the 3D perspective is fixed to a single predefined angle.

A.3.13. Inappropriate Axis Range

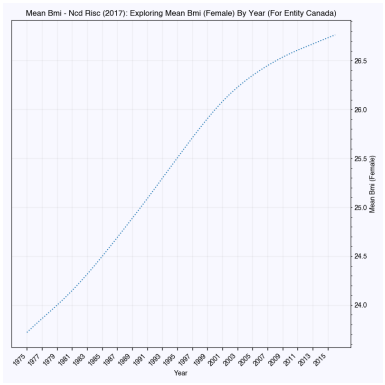


Figure A.17.: Example for a data visualization with an *inappropriate axis range*.

Definition
In the case of an inappropriate axis range, the axis range is either too broad or too narrow to accurately visualize the data, allowing changes to be minimized or maximized depending on the author's intention [5]. Exclusively applied to single axis charts.

Implementation

This misleader is only applied to line charts. In the synthetic dataset, only the case of a too-narrow case is covered, similar to how the *truncated y-axis* is implemented for bar charts, as this is more reliably indicative of misleading intent. By contrast, the case of an overly broad or not truncated y-axis is not included, as its misleading nature is heavily context-dependent and challenging to determine automatically.

For example, consider a line chart showing average daily temperature in a city over the course of a week, ranging narrowly between 19.8°C and 20.2°C. If the y-axis spans from 0°C to 100°C, the resulting line will appear almost flat, potentially downplaying small but meaningful temperature shifts. However, it's unclear whether this is misleading. In some contexts, showing the full 0–100°C range might be justified — for instance, to maintain consistency across a dashboard of climate indicators. In other cases, truncating the axis to focus on the 19–21°C range might better reflect meaningful variability.

The challenge lies in the fact that the misleading factor of a non-truncated (wide-range) axis often depends on the chart's communicative intent, domain conventions, and the viewer's expectations. These factors are challenging to model synthetically in a rule-based way. As a result, this case was excluded from the synthetic data generation, and only the case of a y-axis which is too wide was applied.

Similar to the *truncated y-axis* misleader, the axis is only truncated if it visually exaggerates the differences in the data, under the assumption that a non-misleading version of the chart would use a non-truncated axis starting from zero. The misleader is only applied if the ratio between the minimum and maximum value exceeds 50%, ensuring that the truncation introduces a noticeable distortion.

A.3.14. Dual Axis

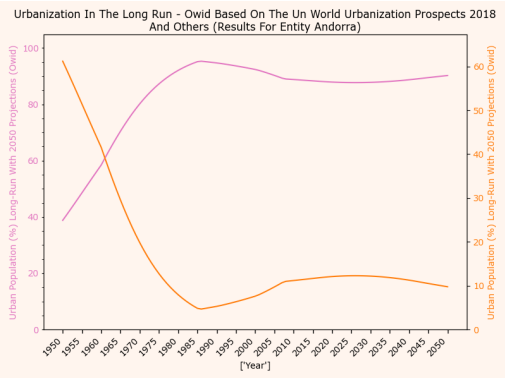


Figure A.18.: Example for a data visualization with the *dual axis* misleader applied.

Definition
Dual axis is when two independent axes are layered on top of each other with inappropriate scaling. This results in a misleading narrative about the relationship between the two [5].

Implementation
This misleader is applied exclusively to line charts using dual axes (both left and right y-axes). Two columns from a multi-column input table are selected. To ensure that the data series displayed on the y-axis are not the same, the y-axis ranges must not overlap by more than 50%, specifically, the overlap must be less than 50% of the smaller range.

This misleader is applied exclusively to line charts using dual y-axes (left and right). Two columns from a multi-column input table are selected. To ensure that the data series scales differ meaningfully, the ranges of the two y-axes must overlap by less than 50% of the smaller range.

A.3.15. Inappropriate Use of Line Chart

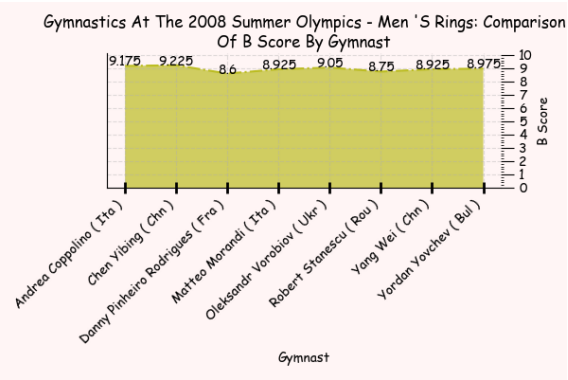


Figure A.19.: An example for an *inappropriate use of a line chart*.

Definition
A line chart is deemed inappropriate when used in an unconventional way or in a way that results in incorrect interpretation of the data or intentionally misleading the audience [5].

Implementation
This misleader is applied exclusively to line charts. It plots a line chart using a categorical independent variable.

A.4. Chart Type Variations

The following chart design variations were incorporated into the *Misviz Synthetic* dataset to increase visual diversity:

All Charts

- Random color variation of the background
- Random title templates
- Random font selection
- Random chart size
- Random font size

Coordinate-Based Charts (Bar and Line Charts)

- Random addition of minor ticks in addition to major ticks on axes

- Random placement of y-axis (left or right side of the chart, *dual axis* and bar charts without y-axis are excluded)
- Different tick shapes
- Multiple series support
- Randomly added value labels (always plotted for bar charts without y-axis)
- Varying tick step size
- Random addition or removal of chart borders
- Random addition of horizontal grid lines

Bar Charts

- Sorted by values or names
- Plots with no y-axis (50% of all bar charts)
- Label placement on top of bars or inside bars
- Horizontal and vertical value labels
- Randomization of bar colors (single random color for time series, multiple random colors in case of a categorical independent variable)

Line Charts

- Randomized fill below the line
- Random line style
- Randomized line colors
- Random addition of horizontal and vertical grid lines

Pie Charts

- Placement of data labels next to slices or in legend

A.5. Additional Metadata *Misviz Synthetic*

Chart Type	Count
BAR_CHART	38.977
LINE_CHART	38.178
PIE_CHART	5.883

Table A.2.: Chart Type Distribution in the *Misviz Synthetic* Dataset.

B. Classification Model Details

B.1. Baseline Prompts

B.1.1. Misviz Dataset Classification Prompt

```
You are a visual chart misleader classification expert. Given an image of a
    ↪chart, identify the correct misleader in the chart from the options
    ↪below.

- NO_MISLEADER: A chart without any misleaders.
- TRUNCATED_Y_AXIS: In a truncated axis, the axis does not start from zero
    ↪or is truncated in the middle, resulting in an exaggerated difference
    ↪between bars of different values.
- INVERTED_AXIS: An axis is oriented in an unconventional direction and the
    ↪perception of the data is reversed
- INAPPROPRIATE_ITEM_ORDER: Items are arranged in an unconventional order,
    ↪misleading the audience or creating confusion.
- INCONSISTENT_INTERVALS: Inconsistent axis ticks refer to cases with
    ↪varying intervals between the ticks.
- THREE_D: A chart is plotted in 3D, making data representations appear
    ↪distorted. For 3D, the closer something is, the larger it appears,
    ↪despite being the same size in 3D perspective.
- MISREPRESENTATION: Misrepresentation occurs when the value labels provided
    ↪do not match the visual encoding. For example, the data values may be
    ↪drawn disproportionately or not to scale, thus intentionally or
    ↪accidentally to cause the data to be misrepresented.
- INAPPROPRIATE_BINNING_SIZE: Inconsistent binning size occurs when there
    ↪are variations in the boundaries of the binning groups of a chart with
    ↪data bins.
- INAPPROPRIATE_USE_OF_PIE_CHART: When a pie chart is used for
    ↪non-part-to-whole data, it creates confusion for the audience, who may
    ↪misinterpret the significance of a given section. Thus, the labels do
    ↪not represent the indicated label shown on the pie slice.
- DUAL_AXIS: Dual axis is when two independent axes are layered on top of
    ↪each other with inappropriate scaling. This results in a misleading
    ↪narrative about the relationship between the two. However, other
    ↪misleaders such as inappropriate axis range might apply as well.
```

```
- INAPPROPRIATE_AXIS_RANGE: In the case of an inappropriate axis range, the
    ↪axis range is either too broad or too narrow to accurately visualize the
    ↪data, allowing changes to be minimized or maximized depending on the
    ↪author's intention. This misleader is exclusively applied to single axis
    ↪charts.
- INAPPROPRIATE_USE_OF_LINE_CHART: A line chart is deemed inappropriate when
    ↪used in an unconventional way or in a way that results in incorrect
    ↪interpretation of the data or intentionally misleading the audience.

Respond with a JSON object with the field 'predicted_label'. The
    ↪predicted_label field should just contain a string of one of the above
    ↪mentioned labels. Nothing else.
```

B.1.2. *Misviz Synthetic* Dataset Classification Prompt

This prompt contains the following additional labels:

- INVERTED_X_AXIS
- INVERTED_Y_AXIS
- NON_LINEAR_Y_AXIS
- INAPPROPRIATE_ACCUMULATION

The following misleaders are not included in the prompt:

- INVERTED_AXIS

Prompt:

```
You are a visual chart misleader classification expert. Given an image of a
    ↪chart, identify the correct misleader in the chart from the options
    ↪below.

- NO_MISLEADER: A chart without any misleaders.
- TRUNCATED_Y_AXIS: In a truncated axis, the axis does not start from zero
    ↪or is truncated in the middle, resulting in an exaggerated difference
    ↪between bars of different values.
- INAPPROPRIATE_ITEM_ORDER: Items are arranged in an unconventional order,
    ↪misleading the audience or creating confusion.
- INCONSISTENT_INTERVALS: Inconsistent axis ticks refer to cases with
    ↪varying intervals between the ticks.
- THREE_D: A chart is plotted in 3D, making data representations appear
    ↪distorted. For 3D, the closer something is, the larger it appears,
    ↪despite being the same size in 3D perspective.
- MISREPRESENTATION: Misrepresentation occurs when the value labels provided
    ↪do not match the visual encoding. For example, the data values may be
    ↪drawn disproportionately or not to scale, thus intentionally or
    ↪accidentally to cause the data to be misrepresented.
```

```
- INAPPROPRIATE_BINNING_SIZE: Inconsistent binning size occurs when there
    ↪are variations in the boundaries of the binning groups of a chart with
    ↪data bins.
- INAPPROPRIATE_USE_OF_PIE_CHART: When a pie chart is used for
    ↪non-part-to-whole data, it creates confusion for the audience, who may
    ↪misinterpret the significance of a given section. Thus, the labels do
    ↪not represent the indicated label shown on the pie slice.
- DUAL_AXIS: Dual axis is when two independent axes are layered on top of
    ↪each other with inappropriate scaling. This results in a misleading
    ↪narrative about the relationship between the two. However, other
    ↪misleaders such as inappropriate axis range might apply as well.
- INAPPROPRIATE_AXIS_RANGE: In the case of an inappropriate axis range, the
    ↪axis range is either too broad or too narrow to accurately visualize the
    ↪data, allowing changes to be minimized or maximized depending on the
    ↪author's intention. This misleader is exclusively applied to single axis
    ↪charts.
- INAPPROPRIATE_USE_OF_LINE_CHART: A line chart is deemed inappropriate when
    ↪used in an unconventional way or in a way that results in incorrect
    ↪interpretation of the data or intentionally misleading the audience.
- NON_LINEAR_Y_AXIS: Scaling of the y-axis of the data changes in the middle
    ↪of the axis, leading to a non linear scale, which makes the data after
    ↪the scale change appear to be more or less than it actually is.
- INAPPROPRIATE_ACCUMULATION: A cumulative measure is used to hide a
    ↪declining trend in the data.
- INVERTED_X_AXIS: The x-axis is oriented in an unconventional direction and
    ↪the perception of the data is reversed.
- INVERTED_Y_AXIS: The y-axis is oriented in an unconventional direction and
    ↪the perception of the data is reversed.

Respond with a JSON object with the field 'predicted_label'. The
    ↪predicted_label field should just contain a string of one of the above
    ↪mentioned labels. Nothing else.
```

B.2. Loss and F_1 Score over Epochs

The following presents the training loss curves for the best-performing model from the ablation study and the model trained under realistic input conditions.

For training the models for the real-world generalization experiment, the macro-averaged F_1 is tracked across epochs on the *Misviz* and *Misviz Synthetic* validation datasets (Stage 2 and 3).

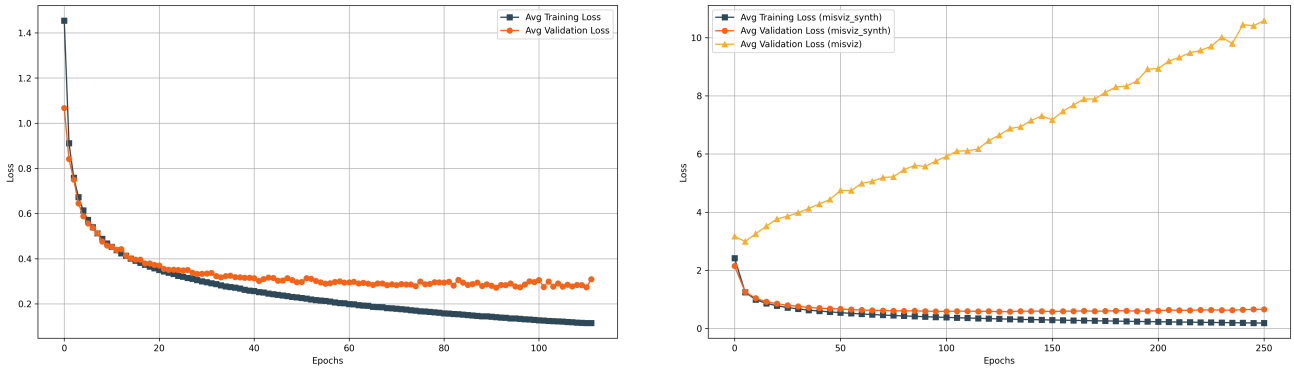


Figure B.1.: Left: Training and validation loss over epochs on *Misviz Synthetic* for the best-performing model from the ablation study (Stage 2). Right: Training and validation loss over epochs on *Misviz Synthetic* and validation loss on *Misviz* for the best-performing classifier on the *Misviz* validation set trained under realistic input conditions (Stage 2).

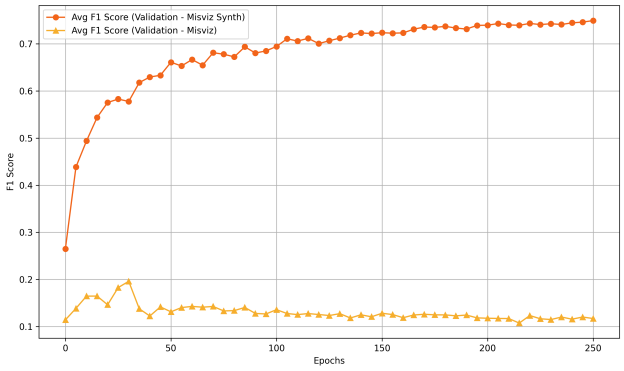


Figure B.2.: Macro-averaged F_1 score across training epochs on the *Misviz Synthetic* and *Misviz* validation sets for the best-performing model selected based on *Misviz* validation set performance.

B.3. *Misviz Synthetic* and *Misviz* Label Mapping

To enable consistent evaluation across both the synthetic and real-world datasets, a mapping is required from the more fine-grained label set of the *Misviz Synthetic* dataset to the coarser label taxonomy used in the *Misviz* real-world dataset. This section outlines the strategy for aligning synthetic misleader categories with their closest equivalents in the real-world classification prompt.

B.3.1. Overview of Label Differences

The *Misviz Synthetic* dataset differs from the real-world label set by introducing several more specific misleader types:

- `INVERTED_X_AXIS`
- `INVERTED_Y_AXIS`

- NON_LINEAR_Y_AXIS
- INAPPROPRIATE_ACCUMULATION

At the same time, the synthetic dataset does not include the generalized label INVERTED_AXIS, which appears in the real-world dataset. A label mapping is applied to translate synthetic labels to their real-world equivalents to ensure compatibility during evaluation.

B.3.2. Label Mapping Strategy

Table B.1 presents the mapping applied from synthetic to real-world labels.

Synthetic Label	Mapped Real-World Label
NO_MISLEADER	NO_MISLEADER
TRUNCATED_Y_AXIS	TRUNCATED_Y_AXIS
INVERTED_X_AXIS	INVERTED_AXIS
INVERTED_Y_AXIS	INVERTED_AXIS
INAPPROPRIATE_ITEM_ORDER	INAPPROPRIATE_ITEM_ORDER
INCONSISTENT_INTERVALS	INCONSISTENT_INTERVALS
THREE_D	THREE_D
MISREPRESENTATION	MISREPRESENTATION
INAPPROPRIATE_BINNING_SIZE	INAPPROPRIATE_BINNING_SIZE
INAPPROPRIATE_USE_OF_PIE_CHART	INAPPROPRIATE_USE_OF_PIE_CHART
DUAL_AXIS	DUAL_AXIS
INAPPROPRIATE_AXIS_RANGE	INAPPROPRIATE_AXIS_RANGE
INAPPROPRIATE_USE_OF_LINE_CHART	INAPPROPRIATE_USE_OF_LINE_CHART
NON_LINEAR_Y_AXIS	— (no equivalent)
INAPPROPRIATE_ACCUMULATION	— (no equivalent)

Table B.1.: Mapping from *Misviz Synthetic* labels to real-world *Misviz* dataset labels.

B.3.3. Mapping Considerations

The real-world label INVERTED_AXIS is used as a general category for both x- and y-axis inversions. Therefore, the synthetic labels INVERTED_X_AXIS and INVERTED_Y_AXIS are mapped to this single class. In contrast, the synthetic dataset contains additional categories such as NON_LINEAR_Y_AXIS and INAPPROPRIATE_ACCUMULATION, which have no equivalent in the real-world label set and are thus excluded from evaluation when comparing across datasets.

B.4. Class-Based F_1 Scores for Best Performing Trained Classifiers

B.4.1. Misviz Synth

Label	F_1 Score
NO_MISLEADER	0.547
TRUNCATED_Y_AXIS	0.559
INVERTED_Y_AXIS	0.899
NON_LINEAR_Y_AXIS	0.798
INAPPROPRIATE_ITEM_ORDER	0.781
INCONSISTENT_INTERVALS	0.593
THREE_D	1.000
MISREPRESENTATION	0.553
INAPPROPRIATE_ACCUMULATION	0.848
INAPPROPRIATE_BINNING_SIZE	0.967
INAPPROPRIATE_USE_OF_PIE_CHART	0.526
DUAL_AXIS	0.983
INAPPROPRIATE_AXIS_RANGE	0.792
INAPPROPRIATE_USE_LINE_CHART	0.976
INVERTED_X_AXIS	0.910
Macro Avg.	**0.782**

Table B.2.: **Training Under Realistic Input Conditions: F_1 scores for each misleader class for the best performing trained model.** F_1 scores per misleader class and overall macro-average F_1 for the best performing model (*TinyChart* vision encoder + *Axis-DePlot* axis metadata extraction) on the test set of *Misviz Synth*.

B.4.2. Misviz

Label	F_1 Score
NO_MISLEADER	0.107
TRUNCATED_Y_AXIS	0.305
INAPPROPRIATE_ITEM_ORDER	0.000
INCONSISTENT_INTERVALS	0.000
THREE_D	0.142
MISREPRESENTATION	0.519
INAPPROPRIATE_BINNING_SIZE	0.182
INAPPROPRIATE_USE_OF_PIE_CHART	0.316
DUAL_AXIS	0.118
INAPPROPRIATE_AXIS_RANGE	0.054
INAPPROPRIATE_USE_LINE_CHART	0.487
INVERTED_AXIS	0.032
Macro Avg.	**0.188**

Table B.3.: **Generalization to Real-World Data: F_1 scores for each misleader class for the best performing trained model.** F_1 scores per misleader class and overall macro-average F_1 for the best performing model (*TinyChart* vision encoder + *Axis-DePlot* axis metadata extraction) on the test set of *Misviz*.

B.5. Binary Classification Results

To assess the performance of the trained models from Stage 2 in a binary classification setting, the results from Stage 2 (training under realistic input conditions) and Stage 3 (generalization to real-world data) were re-aggregated into two classes: instances originally labeled as *no misleader* were retained, while all other misleader types were grouped under the label *contains misleader*. Performance across all classifiers was then compared based on this binary relabeling.

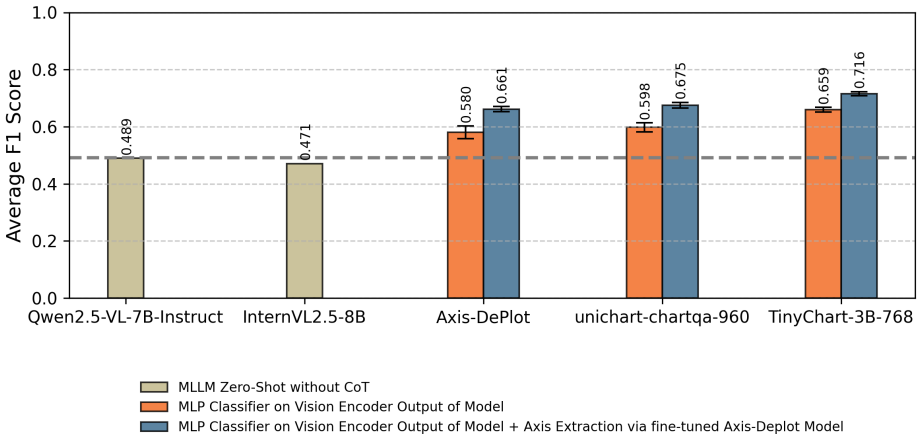


Figure B.3.: Binary classification results on the *Misviz Synthetic* test set.

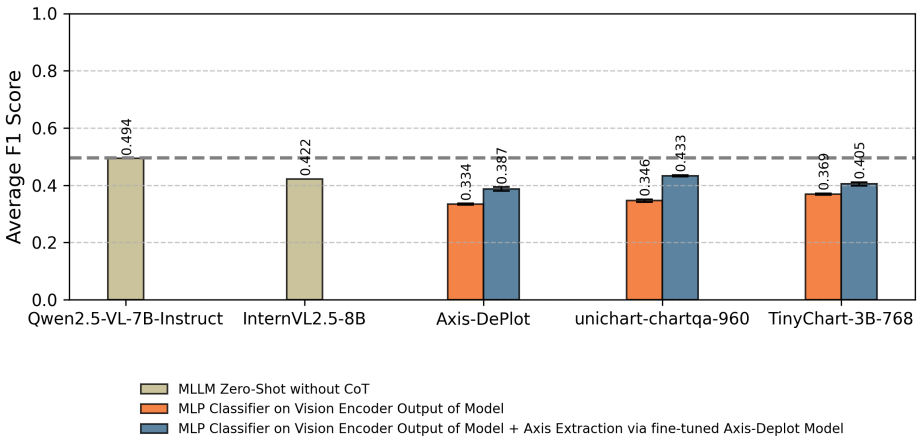


Figure B.4.: Binary classification results on the *Misviz* test set.