Use of Data Visualisations in Software Engineering Research

Meret Baumgartner
Vrije Universiteit Amsterdam, The Netherlands
m.s.baumgartner@student.vu.nl

ABSTRACT

Data visualisations can have a great impact on whether the information in a scientific publication can be successfully conveyed, and are part of nearly all software engineering research papers. In this study, we compile a dataset of 4293 data visualisations and their categorisations, extracted from software engineering conferences' 2024 proceedings. We analyse these data visualisations regarding their position in their respective papers, their chart categories, aspect ratios, colour use and visual density. Based on this analysis and existing visualisation and cognition research, we also formulate recommendations to software engineering researchers on how to improve the use of data visualisation in scientific software engineering publications.

KEYWORDS

visualisations, software engineering research, systematic mapping study

1 INTRODUCTION

Visualisations can be an important part of scientific publications. They increase understanding, enable the authors to highlight important points and provoke further thinking about the topic at hand [19, 46]. In software engineering, visualisations are often used to illustrate processes or architectures and to showcase collected data. There has been some research into the use of visualisations in computer science education [15, 17, 25, 26, 54]. Moreover, the study and development of visualisation tools and methods is an active area of research within the computer science community, with conferences dedicated to the topic such as IEEE VIS[1], IVAPP[2] and EUROVIS[3]. In other fields some evaluations of the use of visualisations in scientific publications have been carried out (e.g. [33, 39]), but also there, research is limited. Datasets of visualisations in scientific publications in the fields of computer vision and visualisations have been collected (e.g. [12]), but as Deng et al. [20] remark, they do not go so far as to analyse the collected images.

The goal of this study is to collect and analyse data visualisations from recently published software engineering papers. By doing so we contribute to the software engineering community by providing a dataset of categorised 5251 Figures extracted from 1046 papers across 17 software engineering conferences. This dataset can form a reference point to software engineering researchers, providing visualisations of various categories and with various properties. We provide an overview of what visualisations are currently used in the field by analysing the visualisations' position in the paper, their categories, and characteristics related to aspect ratio and colour. We then compare these findings to best-practices regarding research visualisations and give suggestions for improvement to the software engineering research community.

The remainder of this paper is structured as follows. Section 2 gives an overview of the relevant literature related to the topics of visualisation in research publications. Section 3 presents the goal of the paper, the research questions we aim to answer, and how the data used was selected, extracted and synthesised. The results of this synthesis are reported in Section 4 and discussed in Section 6, along with some recommendations. Finally, Section 6 closes the paper.

2 RELATED WORK

Aside from visualisations being investigated as a tool in computer science education as stated above, research into visualisations as a tool to support research has been carried out from systems analysis [6, 18], to computer forensics [45], micro-service performance assessments [57], and zero-day malware detection [59]. There is less research on the use of visualisations as a tool for *conveying* research instead of *doing* it, however. [12, 14, 20, 34, 42] collect datasets of visualisations from publications from visualisation and computer vision conferences, but only [20] takes an extra step after this. Next to collecting the visualisations in publications published in the field of visualisation over 22 years, they also analyse these visualisations regarding their types, use case, and position within images. The lack of further research on this topic is particularly striking given the impact properly utilised visualisations can have on the effectiveness of scientific research papers to convey information [44, 60].

Software engineering journals such as *Information and Software Technology* [23] and *Empirical Software Engineering* [55] and publishers more broadly such as Elsevier [22] and IEEE [31, 32] all have submission guidelines, but with regards to figures, these guidelines largely stay on a superficial level, addressing file formats, resolutions, and how to refer to them. They do not address the content of the figures except perhaps to refer to some accessibility guidelines, the most widely used of which are the Web Content Accessibility Guidelines (WCAG) [1]. Although these do not focus on images or figures specifically, they do contain standards that are relevant to producing scientific figures, pertaining to, e.g., colour use. Although there is very little guidance from publishers in the field of software engineering or even computer science more broadly, many guides on how to use visualisations effectively exist (e.g., [10, 35, 38, 58]). It has been noted, however, that visualisation researchers often neglect to take the actual use of these guides into account [47]. Next to such formal guides, informal guides and catalogues exist to help those less experienced navigate the world of visualisation (e.g., [24, 37, 40, 50]).

Finally, there have been attempts at creating taxonomies of visualisation algorithms [56, 62] and the charts they create [4, 7, 11, 52], but as Chen et al. [11] note, this is still an area of development

[1] https://ieeevis.org
[2] https://ivapp.scitevents.org/
[3] https://www.eurovis2025.lu/

where reaching consensus, especially when taking a broad range of visualisations into account, can be very difficult.

3 STUDY DESIGN

The study design for this paper is based on established guidelines for systematic mapping studies [63]. First, we define the research goal (Section 3.1), followed by the research questions (Section 3.2). Then, we search for and select software engineering conferences from which the papers will be used to extract the data (Section 3.3). A full replication package containing the data used for this study together with the scripts used to obtain it can be found on GitHub[4]. It includes all the raw data obtained throughout the study, and the scripts used obtain the data and carry out the analysis.

3.1 Research Goal

The goal of this paper is to characterise the current use of data visualisations by software engineering researchers. This goal is defined in Table 1 according to the Goal, Question, Metric-framework [3].

Purpose	Characterise
Issue	the current use of
Object	data visualisations
Viewpoint	from software engineering researchers' point of view.

Table 1: GQM Research goal definition.

3.2 Research Questions

In order to address the goal defined above, this study aims to address the following research questions:

- **RQ1** - What are the properties of data visualisations currently being used most in software engineering research? In order to answer RQ1, we further refine it in into the following three research questions.
  - **RQ1.1** - Where are data visualisations used in software engineering research papers?
  - **RQ1.2** - What are the chart categories used most in software engineering research papers?
  - **RQ1.3** - What are the properties of data visualisations used in software engineering research papers?
- **RQ2** - What recommendations for the use of data visualisations by software engineering researchers can we formulate based on the current use of data visualisations in the field?

By answering RQ1, we provide a detailed overview of what visualisations are being used by software engineers. We do so by investigating the position of visualisations in their respective papers in order address RQ1.1, by categorising the visualisation to answer RQ1.2 and by looking at properties such as aspect ratio, colour use and visual density for RQ1.3. By answering RQ2 we support members of the software engineering research community by providing with suggestions on how to utilise data visualisations to their fullest potential in scientific publications.

[4]https://github.com/Meret6832/vis-study

3.3 Search and Selection

The search and selection process performed for this research is visualised in Figure 1. It consists of 5 stages:

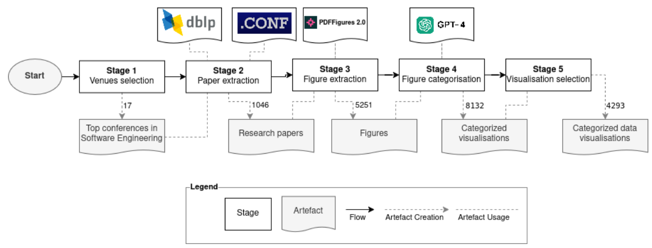


Figure 1: Search and selection process.

3.3.1 Stage 1 - Venues selection. To ensure that the papers used for this research are of high quality and representative of the software engineering research community, the conferences in the CORE research field 4612 (Software engineering) with a CORE 2023 rank of A or A* are gathered[5]. Then, three software engineering researchers each indicate for every conference whether they judge it to be strictly relevant to the software engineering domain. If a conference has been marked as relevant by at least one of the researchers, it is selected. This results in 17 conferences being selected (see Table 2).

3.3.2 Stage 2 - Paper Extraction. In this stage, the 1046 Research Track papers of the selected conferences' 2024 editions are downloaded. In order to do so, a list with the titles of the Research Track papers are scraped from a conf.researchr.org[6]. These are then cross-referenced with a list of titles extracted from the conference proceedings' DBLP[7] page in order to obtain their citations in bibtex format, which are then used to download the PDFs of those papers via the publishers' website.

3.3.3 Stage 3 - Figure Extraction. Once the PDFs of the selected papers have been downloaded, the figures in these papers are extracted using *PDFFigures 2.0*, which is a tool for extracting figures and tables focused on the domain of computer science [16]. In total, 5251 figures were extracted. We manually checked the extracted figures from 17 papers, one randomly selected from each conference, containing 97 figures in total, and found that 91.67% of figures in the same was indeed extracted.

3.3.4 Stage 4 - Figure Categorisation. With the figures having been extracted, the next step is to classify them. Figures are first categorised into two types: data or process visualisations. Here, we define data visualisations as displaying measured data in a visual way. This type contains what would typically be considered charts or graphs, such as bar charts, line graphs and density plots. Process visualisations do not represent measured (numerical) data, but instead show (part of) a process such as a flowchart, a system diagram or even a screenshot. According to these definitions of visualisation types, the figures are then classified using OpenAI's

[5]https://portal.core.edu.au/conf-ranks/?search=4612&by=for&source=CORE2023&sort=arank&page=1
[6]https://conf.researchr.org/
[7]https://dblp.org/

gpt-4o-2024-08-06 model[8], which, after some post processing, assigned (partly) invalid categorisations to 3.3% of figures. In order to evaluate the accuracy of the classification of the remaining 96.7%, we randomly selected 215 visualisations from 187 papers and manually categorised them. We found that 5.6% of visualisations had been assigned an incorrect figure type, 41.7% of which were data visualisation figures that were categorised as process visualisations and 58.3% of which were process visualisations that were categorised as data visualisations.

3.3.5 Stage 5 - Visualisation Selection. As shown in Figure 2, a majority of figures in the dataset contain process visualisations. It is notable that there is a big difference between conferences in this regard, however, with the fraction of figures consisting of the data visualisations making up 15.8% to 61.8% of figures in a conference's proceedings. Chen et al. [11] note that process visualisations are particularly difficult to categorise. The goal of this study is not to develop a new taxonomy or method of classification for visualisations, but rather to analyse visualisations with the help of such taxonomies and methods, which is why we chose to focus on the 4293 extracted data visualisations for this study.

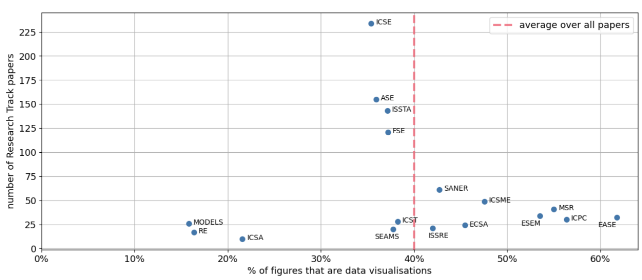


Figure 2: Percentage of figures that are data visualisations and number of Research Track papers for each conference, as well as the average percentage of data visualisation figures.

3.4 Data Extraction

Choosing to focus on data visualisations enabled us to use the taxonomy defined in [7], which is also focused on data visualisations, to categorise them. The taxonomy classifies visualisations according to their chart type, dimensionality and multiplicity. With this taxonomy of data visualisations, gpt-4o-2024-08-06 was again used to categorise the data visualisations in the data set. The same sample of 215 visualisations that was created to obtain an indication of the accuracy of assigning types to visualisations in the previous stage were also manually classified in this stage to determine the accuracy of this classification method. 92.6% of the randomly selected visualisations were completely correctly classified. 96.7% of visualisations was assigned a correct category, 99.5% was assigned a correct subcategory and 99.5% was assigned a correct subsubcategory. All data visualisations in the set were assigned the correct dimensionality, 98.1% was assigned the correct multiplicity and all correctly categorised multipanel figures were assigned the correct number of panels.

Next to the classification of the extracted figures, their first mention and the heading it falls under are extracted. In order to assess the accuracy of the used method for this, 50 figures from 47 papers were randomly selected. For 96% of figures, the extracted page of the first mention was correct, while for only 68% the extracted heading was correct. For 20% of figures no heading was found and for the remaining 12% an incorrect heading was extracted.

Finally, image properties such as height, width and colours used were extracted from each figure using simple pixel-counting methods.

3.5 Data Synthesis

After having extracted the visualisations and their various properties, we carry out a narrative synthesis [49]. This involves describing the characteristics of the primary objects, in this case the data visualisations, in order to gather insights that can help towards answering our research questions. The results of this synthesis are presented in Section 4.

3.6 Threats to Validity

In this section, we discuss potential threats to the validity of this study as well as the actions taken to mitigate them.

3.6.1 External validity. The main external threat to this study is that the papers from which the figures in the dataset were extracted are not representative of the current state of software engineering research. In order to avoid this threat, we looked at a relatively large number of software engineering conferences identified by three software engineering researchers and chose to include a conference even if only one of the researchers indicated it was relevant. Furthermore, we automatically extracted the PDFs of conferences' research papers, thereby minimising human error. By only looking at papers published in conference proceedings and not, e.g., journals, we limited the scope of the study. This also enabled us to look at the *current* state of visualisations in software engineering papers, however, as the submission and publishing process for journals tends to be significantly longer than that for conference proceedings. We safeguarded the quality of the papers selected by only selecting conferences with a CORE ranking of A or A*.

3.6.2 Internal validity. The biggest threat to the internal validity to this study are the methods used to extract and classify the visualisations from the selected papers. We carried out some manual classifications ourselves in order to compare them to the automatically obtained classifications with the methods used in the study and thereby gain some insight into these methods' accuracy. It is important to note, however, that the samples that were manually classified are relatively small, which may impact the accuracy of these accuracy metrics. Extraction of figures was done with an accuracy of approximately 92%. The categorisation of figures into types and of visualisations according to Borkin et al.'s [7] taxonomy was done with gpt-4o-2024-08-06. This is a commercial general large language model that has not been specifically trained for these tasks. In order to increase the accuracy of classifications obtained

[8]https://platform.openai.com/docs/models/gpt-4o

using this model, however, the model was queried three consecutive times with the outputs being post-processed after each query and the figures for which invalid or contradictory outputs were generated being included in the next query iteration. With this method we were able to categorise 97% of figures in the dataset with an accuracy of approximately 94% for the figure type and 92% for the classification of data visualisations. The method that is most error-prone was the extraction of figures' first mentions. The page of the first mention could be successfully extracted for with an accuracy of approximately 96%, but extraction of the mention heading only achieved an accuracy of approximately 68%. A contributing factor to this was no heading being extracted for 20% of figures, but this still leaves 12% of headings being incorrectly assigned. While the automatic methods of classifications have some issues with accuracy, they enabled us to carry out an analysis on a large dataset, something which would not have been possible at the same scale if done manually.

3.6.3 Construct validity. The choice to select a relatively broad range of software engineering conferences and to subsequently select all of their Research Track papers was made to keep the initial selection of source papers broad. To categorise the data visualisations extracted from these papers, we used the taxonomy defined by Borkin et al. [7]. While this taxonomy is not all-encompassing and includes categories for visualisations that are very unlikely to be related to data (e.g., Text-Based), it was created based on a large dataset of visualisations from various sources including scientific publications and its categories largely align with those used in other visualisation research. Other metrics used in this study such as aspect ratio and visual density are metrics that have been proposed and used to evaluate visualisations in other studies, but we are aware that they are not the only metrics that could be used to describe and evaluate visualisations. They enable us to describe the data visualisations in the dataset using a range of different approaches, however, therefore making it possible to sketch a picture of the current use of data visualisations in software engineering research and formulate some recommendations, which is the goal of this study.

3.6.4 Conclusion validity. The methods and metrics used in this study may not be well-suited for an in-depth analysis of the visualisations in the dataset, but this is also not what we aim to achieve. Instead, we aim to give an overview, and are able to do so by performing a higher-level analysis of the dataset of data visualisations. Moreover, in order to formulate recommendations regarding data visualisations in software engineering research, we do not just rely on our own analysis, but also on findings from visualisation and cognition research.

4 RESULTS

In this section, we present the results of the analysis we carried out. In Section 4.1 we address RQ1.1 by discussing the position of figures in their respective papers and the positions and headings of their first mention. Next, for RQ1.2, we investigate the categories of the figures in the dataset in Section 4.2. Finally, in Section 4.3 we address RQ1.3; we look at the physical properties of the figures in the dataset such as aspect ratio, use of colours, and visual density.

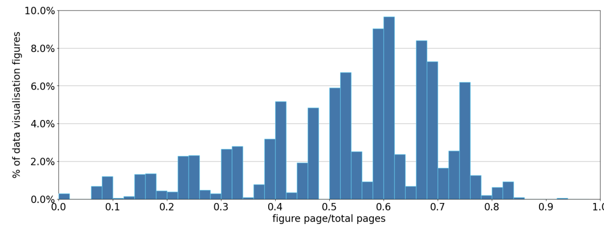


Figure 3: Histogram of the relative position on of data visualisation figures in their paper.

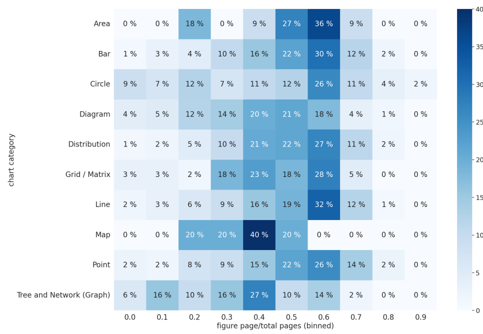


Figure 4: Distribution of figures with a certain chart type over relative positions (binned), excluding chart category *Text Based*.

4.1 Figure position

On average, the number of figures containing data visualisations increases from the beginning to the end of a paper, with most data visualisations being situated between the middle and three-quarter point of a paper (see Figure 3). Most chart categories follow this trend, with *Area*, *Grid / Matrix*, *Map* and *Tree and Network (Graph)* being notable exceptions, most which display spatial relationships (see Figure 4).

These findings about the position of figures also aligns with the heading under which figures' mentions are positioned. By manually clustering headings into section categories, we can see that the biggest group of figures has a first mention under a heading related to *Results* or *Evaluation*, followed by a much smaller group of figures that are placed under headings related to *Methodology* or *Introduction/Background* (see Figure 5).

In Figure 6 we can see that most figures are placed close to their first mention with most figures being first referred to on the same page the figure is positioned or one page after it.

4.2 Figure categorisation

4.2.1 Chart categories. The distribution of chart categories of the data visualisations in the data set is shown in Figure 7. There, we can see that Bar, Distribution and Line visualisations form the biggest categories, followed by Point visualisations. In Figure 8a, we can see that Line Graph visualisations make up 30% of all visualisations. In the Bar category, Grouped Bar Charts form the majority, while this is Box-and-Whisker Plots for the Distribution category. This includes multiple visualisations in one figure. If we look at figures

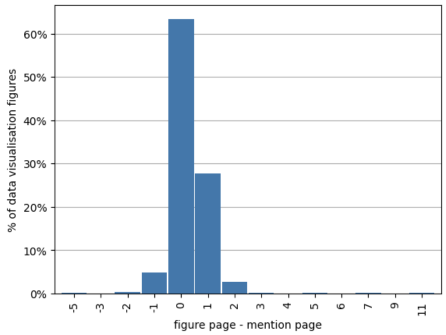


Figure 5: Relative position of the first mention of data visualisation figures.

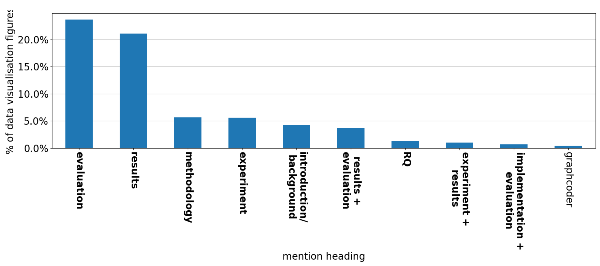


Figure 6: 10 most occuring headings under which data visualisation figures are first referenced. Headings in bold contain multiple headings that have been manually clustered together.

with multiple visualisations and only count them once, this distribution changes (see Figure 8a) and figures in the Bar category now form the biggest group. From this, we can conclude that Line graphs often occur in figures with multiple visualisations, which is supported by Figure 9 which displays the distribution of chart categories for Multipanel figures.

4.2.2 Dimensionality and multiplicity. 99.7% of the data visualisations in the dataset is two-dimensional. The remaining 0.3% (13 visualisations) are three-dimensional.

Most figures in the dataset are Single or Multipanel, as can be seen in Figure 10. Multipanel figures most often consist of 2, 3 or 4 visualisation panels. Of the remaining multiplicity categories, more visualisations fall in the Grouped category, which includes figures that have multiple categories within one visualisation. The smallest number of figures fall in the Combination category, where a figure consists of one visualisation containing multiple different chart categories.

4.3 Image properties

4.3.1 Aspect ratio. As shown in Figure 11, most visualisations from the selected conferences' Research Track papers have a horizontal orientation, meaning that their width is greater than their height. Ratios between 3:2 and 4:3 are the most common.

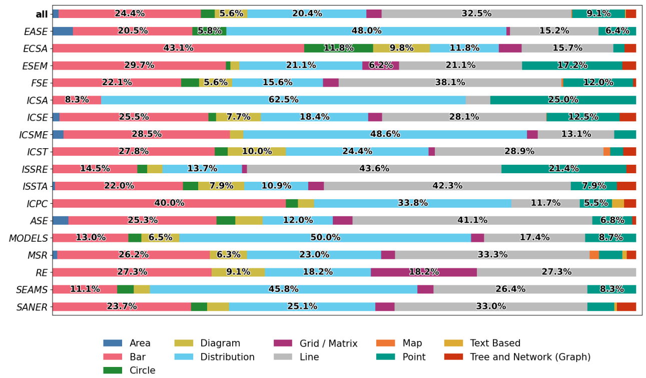
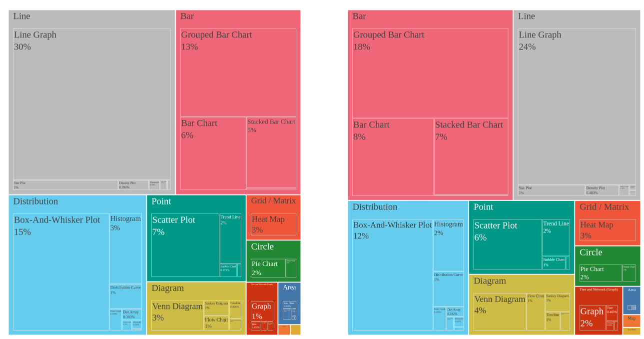


Figure 7: Distribution of chart categories over the whole dataset (all), and for the different conferences. While the thickness of the bar for each conference is the same, not all conferences contributed the same number of papers to the dataset (see Table 2)



(a) Distribution over all data visualisation figures in the dataset.

(b) Distribution over all data visualisations in the dataset.

Figure 8: Distribution of chart categories and subcategories.

4.3.2 Colour use. A large majority of the figures in the data set are in colour (see Figure 12). As can be seen in Figure 13a, the most recurring colour is white. This is most likely because it is the background colour for most figures. White is followed as the most occurring colour by black and other shades of grey. In Figure 13b, shades of grey have been removed and the occurrence of other colours can be better observed. We can see that while certain shades of blue and red occur most often, overall blue is most prominent, followed by green and orange and finally red.

4.3.3 Visual Density. Visual density refers to the overall density of elements in a visualisation, not distinguishing between different kinds of elements such as foreground versus background or chart elements versus text [7]. Here, we refer to visual density as the fraction of images that is not white. A figure with low visual density has very few non-white pixels, while a figure with high visual density has relatively few white pixels. The average visual density across non-Multipanel figures in the dataset is 0.24. The average visual density for each chart category is displayed in Figure Figure 14. There, we can see that figures in the *Distribution*, *Line* and *Point* have below-average visual densities, while *Grid / Matrix* figures have very high visual densities, followed by those in the *Circle* category.

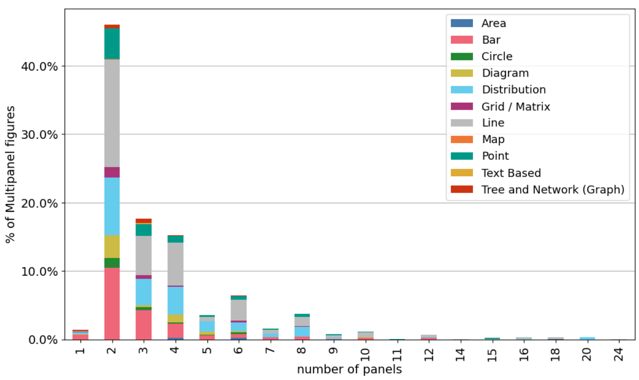


Figure 9: Distribution of the number of panels for Multipanel figures, with the chart category of the Multipanel figures.

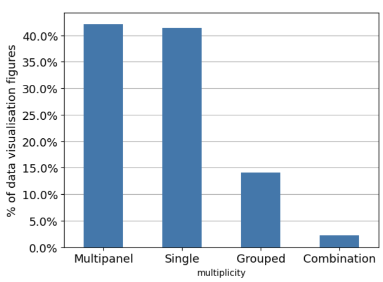


Figure 10: Distribution of different multiplicities for data visualisation figures in the dataset.

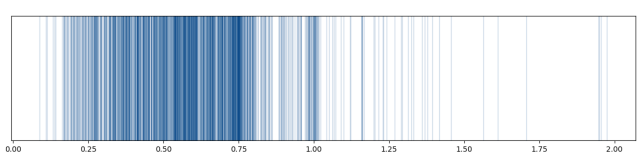


Figure 11: Aspect ratios of the data visualisation figures in the dataset, excluding Multipanel figures. Each vertical line represents one figure.

5 DISCUSSION

In this section we reflect on the results presented in Section 4, position them in the context of visualisation research and formulate recommendations for software engineering researchers using visualisations in their papers. What visualisations are best heavily

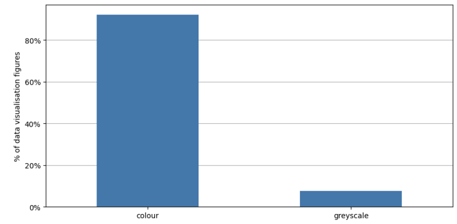
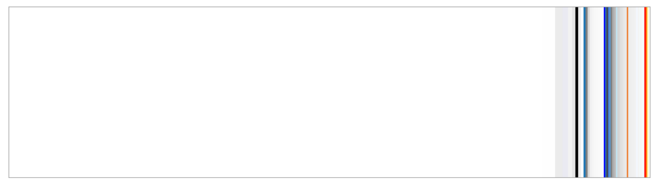
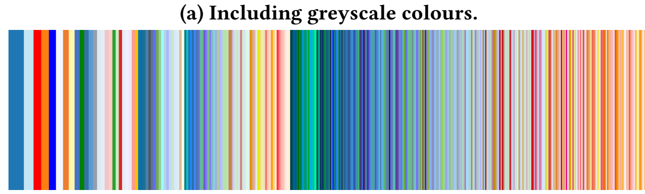


Figure 12: Colour scale of the dataset visualisation figures in the dataset.



(a) Including greyscale colours.



(b) Excluding greyscale colours.

Figure 13: Distribution of colours in all data visualisation figures in the dataset.

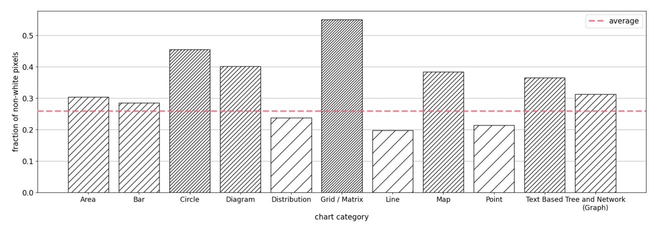


Figure 14: Fraction of non-white pixels for different chart categories.

depends on the context [29], but some general findings and best practices can be identified across scientific fields, which is what we base the recommendations given in this section on.

5.1 Figure position

Most figures in the data set are on the same page as their first mention, or on the page after. This follows general guidelines on figure placement in scientific publications, such as has been formalised by, e.g., IEEE [32]. As we are focusing on data visualisations for this study, it could be expected that most visualisations are placed under a heading related to evaluation or results. There is also a significant number of visualisations that is not placed under such a heading. Figure 15, for example is located under *Related Work*. Figure 16 and 17 are placed under *Background* and *Introduction* headings, and are used to visualise preliminary findings in their respective papers. The dataset presented with this study can form a source of inspiration for researchers to use visualisations in ways that they may not have previously thought of.

5.2 Figure categorisation

5.2.1 Chart categories. The type of chart that is used has a significant impact on how a chart will be interpreted by a reader and with what accuracy. What chart category is best suited to a certain context mainly depends on the goal of the visualisation as this influences both the speed and accuracy of cognition [28]. Perkhofer et al. identify three tasks, compare, identify and summarise. Overall, they find that visualisations are better suited to support summarising than identification. Furthermore, visualisations showing data in a disaggregated form perform better in an identification context, while those presenting data in an aggregated form present better in a summarising context. Regarding specific chart types, statistician Edward Tufte famously wrote that "the only design worse than a pie chart is several of them" [58]. Research does indeed support this, with pie charts being the type of chart that is most often misinterpreted [44], and Cartesian visualisations (which include line charts) being understood with a higher accuracy and speed than radial visualisations (which include pie charts) [9, 48].

Indeed, circle charts are only used sparingly in the papers selected for this study, and circle charts only form 2.4% of all data visualisations. Nguyen et al. [44] also found that, after pie charts, bar charts are the most misinterpreted visualisations. This is especially the case for non-stacked bar charts used to visualise percentages or fractions. In the dataset, *Grouped Bar Charts* (13.3%) and *Bar Charts* (6.1%) together form 19.4% of all visualisations. If we include *Histograms*, this is even 22.5%. While there are certainly contexts in which bar charts are a good solution to visualise data, this is a particularly high percentage for the second-most misinterpreted visualisation category.

> **Recommendation 1**: Only use radial visualisations and bar charts if no other visualisation category forms an alternative.

These insights lead us to formulate *Recommendation 1*, but we do acknowledge that it can be hard to avoid using bar charts, we even use them in this study as well. They are often a default option and many of the alternatives are radial visualisations, which should also be avoided.

Next to looking at the most-used chart categories, we can also pay attention to categories that are less prevalent in the dataset, as these categories may sometimes be a better choice than the "default" categories that are more often used. Examples of such categories are *Bubble Charts* (see, e.g., Figure 19), *Dot Arrays* (see, e.g., Figure 20), *Dot Plots* (see, e.g., Figure 21) and *Slopegraphs* (see, e.g., Figure 22).

5.2.2 Dimensionality and Multiplicity. In general, for visualisations the rule *simpler is better* holds. This is also true for the numbers of dimensions and multiplicity of a visualisation [28, 35, 58]. In the dataset compiled for this study, only 0.3% of figures are three-dimensional. This could indicate that researchers only use three-dimensional visualisations if absolutely necessary. For examples of three-dimensional visualisations in the dataset, see Figures 17 and 23.

For the multiplicity of data visualisations, the same rule holds. Too many panels in a multipanel figure can be overwhelming and combining different chart types in a Combination visualisation figure can make it too cluttered. The visualisations in the dataset perform quite well in this regard, however, with most Multipanel figures having less than 5 panels and only 2.3% of visualisations being Combinations.

> **Recommendation 2**: Reduce the dimensionality and number of panels of visualisations as much as possible.

5.3 Image Properties

5.3.1 Aspect ratio. The orientation of a visualisation has an effect on how it is perceived, with horizontal charts generally being perceived better [58] and increases in the height of a chart not leading to increased accuracy in most cases [28]. Most visualisations (94.5%) in the dataset are horizontal, which shows that most software engineering researchers already follow *Recommendation 3*. Even with horizontal visualisations, especially those with an aspect ratio closer to 1:1, the height of visualisations can still be reduced, however, without decreasing how accurately it is perceived.

> **Recommendation 3**: Use visualisations with a horizontal layout instead of a vertical one and avoid unnecessarily increasing their height.

5.3.2 Colour use. Some best-practices regarding colour use are already widely known such as taking colour-blindness into account [51], avoiding rainbow colour maps [8] and using continuous colour maps for continuous data [53]. Research has also shown that, if possible, restricting the number of coloured categories in one visualisation to less than 8 works best [44]. Furthermore, if possible, colour should not be the only distinguishing factor, but instead be used alongside different hues, patterns and markers [44]. This is in line with research on the impact of crowdedness of visualisations, which has found that they should be kept as simple as possible regarding the fraction of foreground pixels conveying information and as dense packing reduces accuracy in the perception of the visualisation [28, 35, 58].

Only 7.7% of figures in the dataset are greyscale, which could suggest an over-reliance on colour. It is not always the case, however, that because a figure uses some colour, it is doing so without reason. Figures can use colour within a visualisation to make it stand out in the text, or to distinguish between visualisations. As stated above, however, colour should not be the only distinguishing factor. Hue, patterns and symbols can be used to make visualisations clearer and more accessible (see , e.g., Figure 24). When using colours, it is important that the colour combinations used are still distinguishable for colour-blind people and have high contrasts [1].

> **Recommendation 4**: Do not only rely on colours as a distinguishing factor in visualisations, but also utilise symbols, patterns and high contrasts.

5.3.3 Visual density. Visual density is another possible metric to measure the crowdedness of an image. Visualisations with low visual density (e.g., Figure 25) have few non-white pixels and overall pack less information into the visualisation. The visual density metric also has limits. Chart categories that may convey relatively simple information in a way that covers the majority of the visualisation have high densities (e.g., Figures 18 and 26). Furthermore, visualisations with a non-white background also have high visual density (e.g., Figure 27). While a non-white background does not necessarily increase the crowdedness of an image, it does reduce contrast in many cases, which in turn can reduce the readability and accessibility of a visualisation [1].

6 CONCLUSION

In this study, we looked at the current use of data visualisations in software engineering research by analysing a dataset of 4293 data visualisations extracted from 17 software engineering conferences' 2024 proceeding. In order to do so, we classified the images according to their position in their respective papers, their visualisation categories and their physical properties such as aspect ratio, colours and aspect ratio. Based on this analysis and existing research in the field of visual analysis, we also formalised a number of recommendations for software engineering researchers:

- Avoid using radial visualisations and bar charts.
- Reduce the dimensionality and number of panels of visualisations as much as possible.
- Use visualisations with a horizontal layout instead of a vertical one and do not increase their height without needing to do so.
- Do not only rely on colours as a distinguishing factor in visualisations, but also utilise symbols, patterns and high contrasts.

As future work, this study could be taken as a base to further investigate the use of visualisations in software engineering research publications. A more accurate method of classifying images could be developed, for example through using machine learning or manual classification. Instead of looking at visualisations more broadly as done here, they could also be investigated through visual analysis such as content analysis. Furthermore, the use of visualisations in research publications in different contexts could be contrasted, such as over different time periods, between different publishing forms or between different research fields. Finally, studies on the speed and accuracy of visualisations specifically in the software engineering research context, or even computer science more broadly could be carried out.

REFERENCES

[1] 2024. Web Content Accessibility Guidelines (WCAG) 2.2. https://www.w3.org/TR/WCAG22/
[2] Khubaib Amjad Alam, Hira Asif, Irum Inayat, and Saif-Ur-Rehman Khan. 2024. Automated Quality Concerns Extraction from User Stories and Acceptance Criteria for Early Architectural Decisions. In *Software Architecture*, Matthias Galster, Patrizia Scandurra, Tommi Mikkonen, Pablo Oliveira Antonino, Elisa Yumi Nakagawa, and Elena Navarro (Eds.). Springer Nature Switzerland, Cham, 359–367. https://doi.org/10.1007/978-3-031-70797-1_24
[3] Victor R. Basili, Gianluigi Caldiera, and H. Dieter Rombach. 1994. Goal question metric paradigm. In *Encyclopedia of Software Engineering*. Vol. 1. 528–532.
[4] Jacques Bertin. 1983. *Semiology of graphics*. University of Wisconsin Press.
[5] Dirk Beyer, Matthias Kettl, and Thomas Lemberger. 2024. Decomposing Software Verification using Distributed Summary Synthesis. *Proc. ACM Softw. Eng.* 1, FSE (2024), 1307–1329. https://doi.org/10.1145/3660766
[6] Mikael Blomé, Curt R Johansson, and Per Odenrick. 2003. Computer supported visualisation of quality systems developed by network teams. *Applied Ergonomics* 34, 3 (May 2003), 239–247. https://doi.org/10.1016/S0003-6870(03)00035-8
[7] Michelle A. Borkin, Azalea A. Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva, and Hanspeter Pfister. 2013. What Makes a Visualization Memorable? *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (Dec. 2013), 2306–2315. https://doi.org/10.1109/TVCG.2013.234 Conference Name: IEEE Transactions on Visualization and Computer Graphics.
[8] David Borland and Russell M. Taylor Ii. 2007. Rainbow Color Map (Still) Considered Harmful. *IEEE Computer Graphics and Applications* 27, 2 (March 2007), 14–17. https://doi.org/10.1109/MCG.2007.323435 Conference Name: IEEE Computer Graphics and Applications.
[9] Michael Burch and Daniel Weiskopf. 2014. *On the Benefits and Drawbacks of Radial Diagrams.* Springer New York, New York, NY, 429–451. https://doi.org/10.1007/978-1-4614-7485-2_17
[10] John M. Chambers, William S. Cleveland, Beat Kleiner, and Paul A. Tukey. 1983. *Graphical Methods for Data Analysis.* Chapman and Hall/CRC, New York. https://doi.org/10.1201/9781351072304
[11] Jian Chen, Petra Isenberg, Robert S. Laramee, Tobias Isenberg, Michael Sedlmair, Torsten Moeller, and Han-Wei Shen. 2022. Not As Easy As You Think – Experiences and Lessons Learnt from Trying to Create a Bottom-Up Visualization Image Typology. https://doi.org/10.48550/arXiv.2209.07533
[12] Jian Chen, Meng Ling, Rui Li, Petra Isenberg, Tobias Isenberg, Michael Sedlmair, Torsten Möller, Robert S. Laramee, Han-Wei Shen, Katharina Wünsche, and Qiru Wang. 2021. VIS30K: A Collection of Figures and Tables From IEEE Visualization Conference Publications. *IEEE Transactions on Visualization and Computer Graphics* 27, 9 (Sept. 2021), 3826–3833. https://doi.org/10.1109/TVCG.2021.3054916 Conference Name: IEEE Transactions on Visualization and Computer Graphics.
[13] Tao Chen and Miqing Li. 2024. Adapting Multi-objectivized Software Configuration Tuning. *Reproduction package for "Adapting Multi-objectivized Software Configuration Tuning"* 1, FSE (July 2024), 25:539–25:561. https://doi.org/10.1145/3643751
[14] Xi Chen, Wei Zeng, Yanna Lin, Hayder Mahdi AI-maneea, Jonathan Roberts, and Remco Chang. 2021. Composition and Configuration Patterns in Multiple-View Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (Feb. 2021), 1514–1524. https://doi.org/10.1109/TVCG.2020.3030338 Conference Name: IEEE Transactions on Visualization and Computer Graphics.
[15] Daniela Chudá. 2007. Visualization in education of theoretical computer science. In *Proceedings of the 2007 international conference on Computer systems and technologies (CompSysTech '07)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/1330598.1330687
[16] Christopher Clark and Santosh Divvala. 2016. PDFFigures 2.0: Mining Figures from Research Papers. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries (JCDL '16)*. Association for Computing Machinery, New York, NY, USA, 143–152. https://doi.org/10.1145/2910896.2910904
[17] Veronica Cucuiat. 2022. *Exploring students' iterative practices when learning with physical computing kits through data visualisations.* Doctoral. UCL (University College London). https://discovery.ucl.ac.uk/id/eprint/10158757/
[18] Thomas Davidson and Jonathan Mace. 2022. See it to believe it? the role of visualisation in systems research. In *Proceedings of the 13th Symposium on Cloud Computing (SoCC '22)*. Association for Computing Machinery, New York, NY, USA, 419–428. https://doi.org/10.1145/3542929.3563488
[19] Henk W. de Regt. 2014. Visualization as a Tool for Understanding. *Perspectives on Science* 22, 3 (09 2014), 377–396. https://doi.org/10.1162/POSC_a_00139
[20] Dazhen Deng, Yihong Wu, Xinhuan Shu, Jiang Wu, Siwei Fu, Weiwei Cui, and Yingcai Wu. 2023. VisImages: A Fine-Grained Expert-Annotated Visualization Dataset. *IEEE Transactions on Visualization and Computer Graphics* 29, 7 (July 2023), 3298–3311. https://doi.org/10.1109/TVCG.2022.3155440 Conference Name: IEEE Transactions on Visualization and Computer Graphics.
[21] Mouna Dhaouadi, Bentley James Oakes, and Michalis Famelis. 2024. Rationale Dataset and Analysis for the Commit Messages of the Linux Kernel Out-of-Memory Killer. In *Proceedings of the 32nd IEEE/ACM International Conference on Program Comprehension (ICPC '24)*. Association for Computing Machinery, New York, NY, USA, 415–425. https://doi.org/10.1145/3643916.3644413
[22] Elsevier. [n.d.]. Artwork and media instructions | Elsevier policy. https://www.elsevier.com/about/policies-and-standards/author/artwork-and-media-instructions
[23] Elsevier. [n.d.]. Guide for authors - Information and Software Technology - ISSN 0950-5849. https://datavizproject.com/
[24] Ferdio. [n.d.]. Data Viz Project. https://datavizproject.com/
[25] Eric Fouh, Monika Akbar, and Clifford A. Shaffer. 2012. The Role of Visualization in Computer Science Education. *Computers in the Schools* 29, 1-2 (Jan. 2012), 95–117. https://doi.org/10.1080/07380569.2012.651422
[26] Joe Griffin and Julie Pichon. 2010. Seeing is believing: using Data Visualisation for formative feedback in computer supported online learning collaboration.

International Journal of Electronic Business 8, 4-5 (Jan. 2010), 331–341. https://doi.org/10.1504/IJEB.2010.03529 Publisher: Inderscience Publishers.

[27] Yongqi Han, Qingfeng Du, Ying Huang, Jiaqi Wu, Fulong Tian, and Cheng He. 2024. The Potential of One-Shot Failure Root Cause Analysis: Collaboration of the Large Language Model and Small Classifier. In *2024 39th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 931–943. https://doi.org/10.1145/3691620.3695475 ISSN: 2643-1572.

[28] Jeffrey Heer and Michael Bostock. 2010. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. Association for Computing Machinery, New York, NY, USA, 203–212. https://doi.org/10.1145/1753326.1753357

[29] Eric Hehman and Sally Y. Xie. 2021. Doing Better Data Visualization. *Advances in Methods and Practices in Psychological Science* 4, 4 (Oct. 2021), 25152459211045334. https://doi.org/10.1177/25152459211045334 Publisher: SAGE Publications Inc.

[30] Oseremen Joy Idialu, Noble Saji Mathews, Rungroj Maipradit, Joanne M. Atlee, and Mei Nagappan. 2024. Whodunit: Classifying Code as Human Authored or GPT-4 Generated - A case study on CodeChef problems. In *Proceedings of the 21st International Conference on Mining Software Repositories (MSR '24)*. Association for Computing Machinery, New York, NY, USA, 394–406. https://doi.org/10.1145/3643991.3644926

[31] IEEE. 2024. Create Graphics for Your Article. https://journals.ieeeauthorcenter.ieee.org/create-your-ieee-journal-article/create-graphics-for-your-article/

[32] IEEE. 2025. IEEE Editorial Style Manual for Authors. https://docs.google.com/document/d/1OalFYqVfxKKMelF79U3B93SjmIEiTnJC12wPJrKeSDM/edit

[33] Helena Jambor, Alberto Antonietti, Bradly Alicea, Tracy L. Audisio, Susann Auer, Vivek Bhardwaj, Steven J. Burgess, Julia Ferling, Małgorzata Anna Gazda, Luke H. Hoeppner, Vinodh Ilangovan, Hung Lo, Mischa Olson, Salem Yousef Mohamed, Sarvenaz Sarabipour, Aalok Varma, Kaivalya Walavalkar, Erin M. Wissink, and Tracey L. Weissgerber. 2021. Creating clear and informative image-based figures for scientific publications. *PLoS Biology* 19, 3 (2021). https://doi.org/10.1371/journal.pbio.3001161

[34] K. V. Jobin, Ajoy Mondal, and C. V. Jawahar. 2019. DocFigure: A Dataset for Scientific Document Figure Classification. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, Vol. 1. 74–79. https://doi.org/10.1109/ICDARW.2019.00018

[35] Christa Kelleher and Thorsten Wagener. 2011. Ten guidelines for effective data visualization in scientific publications. *Environ. Model. Softw.* 26, 6 (June 2011), 822–827. https://doi.org/10.1016/j.envsoft.2010.12.006

[36] Kisub Kim, Jounghoon Kim, Byeongjo Park, Dongsun Kim, Chun Yong Chong, Yuan Wang, Tiezhu Sun, Daniel Tang, Jacques Klein, and Tegawende F. Bissyande. 2024. DataRecipe — How to Cook the Data for CodeLLM?. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering (ASE '24)*. Association for Computing Machinery, New York, NY, USA, 1206–1218. https://doi.org/10.1145/3691620.3695593

[37] Andy Kirk. [n.d.]. The Chartmaker Directory. https://chartmaker.visualisingdata.com/

[38] Andy Kirk. 2019. *Data Visualisation: A Handbook for Data Driven Design* (second ed.). SAGE Publications Ltd.

[39] Po-Shen Lee, Jevin D. West, and Bill Howe. 2018. Viziometrics: Analyzing Visual Information in the Scientific Literature. *IEEE Transactions on Big Data* 4, 1 (March 2018), 117–129. https://doi.org/10.1109/TBDATA.2017.2689038 Conference Name: IEEE Transactions on Big Data.

[40] Chenxin Li. 2024. Friends Don't Let Friends Make Bad Graphs. (10 December 2024). https://doi.org/10.5281/zenodo.7542491

[41] Yunqi Liu and Wei Song. 2024. FunRedisp: Reordering Function Dispatch in Smart Contract to Reduce Invocation Gas Fees. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2024)*. Association for Computing Machinery, New York, NY, USA, 516–527. https://doi.org/10.1145/3650212.3652146

[42] Anita Mahinpei, Zona Kostic, and Chris Tanner. 2022. LineCap: Line Charts for Data Visualization Captioning Models. In *2022 IEEE Visualization and Visual Analytics (VIS)*. 35–39. https://doi.org/10.1109/VIS54862.2022.00016 ISSN: 2771-9553.

[43] Gustave Monce, Thomas Couturou, Yasmine Hamdaoui, Thomas Degueule, and Jean-Rémy Falleri. 2024. Lightweight Syntactic API Usage Analysis with UCov. In *Proceedings of the 32nd IEEE/ACM International Conference on Program Comprehension (ICPC '24)*. Association for Computing Machinery, New York, NY, USA, 426–437. https://doi.org/10.1145/3643916.3644415

[44] Vinh T. Nguyen, Kwanghee Jung, and Vibhuti Gupta. 2021. Examining data visualization pitfalls in scientific publications. *Visual Computing for Industry, Biomedicine, and Art* 4, 1 (Oct. 2021), 27. https://doi.org/10.1186/s42492-021-00092-y

[45] Grant Osborne and Benjamin Turnbull. 2009. Enhancing Computer Forensics Investigation through Visualisation and Data Exploitation. In *2009 International Conference on Availability, Reliability and Security*. 1012–1017. https://doi.org/10.1109/ARES.2009.120

[46] Marilyn Ostergren. 2013. *How Scientists Develop Competence in Visual Communication.* PhD Dissertation. University of Washington. https://digital.lib.washington.edu/researchworks/bitstream/handle/1773/23658/Ostergren_washington_0250E_11729.pdf Dissertation.

[47] Paul Parsons and Prakash Shukla. 2022. Considering the Role of Guidelines in Visualization Design Practice. https://doi.org/10.31219/osf.io/mw376

[48] Lisa Perkhofer, Conny Walchshofer, and Peter Hofer. 2020. Does design matter when visualizing Big Data? An empirical study to investigate the effect of visualization type and interaction use. *Journal of Management Control* 31, 1 (2020), 55–95. https://doi.org/10.1007/s00187-020-00294-0

[49] Jennie Popay, Helen Roberts, Amanda Sowden, Mark Petticrew, Lisa Arai, Mark Rodgers, Nicky Britten, Katrina Roen, and Steven Duffy. 2006. *Guidance on the conduct of narrative synthesis in systematic reviews: A product from the ESRC Methods Programme*. https://doi.org/10.13140/2.1.1018.4643

[50] Severino Ribecca. 2024. The Data Visualisation Catalogue. https://datavizcatalogue.com/index.html

[51] Duccio Rocchini, Ludovico Chieffallo, Elisa Thouverai, Rossella D'Introno, Francesca Dagostin, Emma Donini, Giles Foody, Simon Garnier, Guilherme G. Mazzochini, Vitezslav Moudry, Bob Rudis, Petra Simova, Michele Torresani, and Jakub Nowosad. 2024. Under the mantra: 'Make use of colorblind friendly graphs'. *Environmetrics* 35, 6 (2024), e2877. https://doi.org/10.1002/env.2877 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/env.2877.

[52] B. Shneiderman. 1996. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*. 336–343. https://doi.org/10.1109/VL.1996.545307

[53] Samuel Silva, Joaquim Madeira, and Beatriz Sousa Santos. 2007. There is More to Color Scales than Meets the Eye: A Review on the Use of Color in Visualization. In *2007 11th International Conference Information Visualization (IV '07)*. 943–950. https://doi.org/10.1109/IV.2007.113 ISSN: 1550-6037.

[54] Slavomír Simoňák. 2016. Algorithm visualizations as a way of increasing the quality in computer science education. In *2016 IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*. 153–157. https://doi.org/10.1109/SAMI.2016.7422999

[55] Springer. [n.d.]. Submission guidelines | Empirical Software Engineering. https://link.springer.com/journal/10664/submission-guidelines

[56] M. Tory and T. Moller. 2004. Rethinking Visualization: A High-Level Taxonomy. In *IEEE Symposium on Information Visualization*. 151–158. https://doi.org/10.1109/INFVIS.2004.59

[57] Luca Traini, Jessica Leone, Giovanni Stilo, and Antinisca Di Marco. 2024. VAMP: Visual Analytics for Microservices Performance. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing (SAC '24)*. Association for Computing Machinery, New York, NY, USA, 1209–1218. https://doi.org/10.1145/3605098.3636069

[58] Edward R. Tufte. 1986. *The visual display of quantitative information* (second ed.). Graphics Press, USA.

[59] Sitalakshmi Venkatraman and Mamoun Alazab. 2018. Use of Data Visualisation for Zero-Day Malware Detection. *Security and Communication Networks* 2018, 1 (2018), 1728303. https://doi.org/10.1155/2018/1728303 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2018/1728303.

[60] Chaoli Wang and Jun Tao. 2017. Graphs in Scientific Visualization: A Survey. *Computer Graphics Forum* 36, 1 (Feb. 2017), 263–287. https://doi.org/10.1111/cgf.12800 Publisher: John Wiley & Sons, Ltd.

[61] Thomas Weber, Christina Winiker, and Sven Mayer. 2024. An Investigation of How Software Developers Read Machine Learning Code. In *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '24)*. Association for Computing Machinery, New York, NY, USA, 165–176. https://doi.org/10.1145/3674805.3686678

[62] Leland Wilkinson. 2005. *The grammar of graphics* (second ed.). Springer-Verlag, Berlin, Heidelberg. https://doi.org/10.1007/0-387-28695-0

[63] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Bjö Regnell, and Anders Wesslén. 2024. Systematic Literature Studies. In *Experimentation in software engineering* (second edition ed.). Springer, Berlin, Heidelberg. doi: 10.1007/978-3-662-69306-3.

[64] Sara Yabesi, Mahta Amini, Jelena Ristic, and Zohreh Sharafi. 2024. Exploring the Effects of Urgency and Reputation in Code Review: An Eye-Tracking Study. In *Proceedings of the 32nd IEEE/ACM International Conference on Program Comprehension (ICPC '24)*. Association for Computing Machinery, New York, NY, USA, 202–213. https://doi.org/10.1145/3643916.3644425

[65] Shasha Zhou, Mingyu Huang, Yanan Sun, and Ke Li. 2024. Evolutionary Multi-objective Optimization for Contextual Adversarial Example Generation. *Proc. ACM Softw. Eng.* 1, FSE (July 2024), 101:2285–101:2308. https://doi.org/10.1145/3660808

A APPENDIX

Acronym	Title	CORE ranking[9]	Research Track Papers
ASE	Automated Software Engineering Conference	A*	155
FSE	ACM International Conference on the Foundations of Software Engineering	A*	121
ICSE	International Conference on Software Engineering	A*	234
EASE	International Conference on Evaluation and Assessment in Software Engineering	A	32
ECSA	European Conference on Software Architecture	A	24
ESEM	International Symposium on Empirical Software Engineering and Measurement	A	34
ICPC	IEEE International Conference on Program Comprehension	A	30
ICSA	International Conference on Software Architecture	A	10
ICSME	IEEE International Conference on Software Maintenance and Evolution	A	49
ICST	International Conference on Software Testing, Verification and Validation	A	28
ISSRE	International Symposium on Software Reliability Engineering	A	21
ISSTA	International Symposium on Software Testing and Analysis	A	143
MODELS	International Conference on Model Driven Engineering Languages and Systems	A	26
MSR	IEEE International Working Conference on Mining Software Repositories	A	41
RE	IEEE International Requirements Engineering Conference	A	17
SANER	IEEE International Conference on Software Analysis, Evolution and Reengineering	A	61
SEAMS	International Symposium on Software Engineering for Adaptive and Self-Managing Systems	A	20

Table 2: The selected conferences, their CORE ranking and number of extracted Research Track papers in the dataset.

[9]https://portal.core.edu.au/conf-ranks/

A.1 Examples

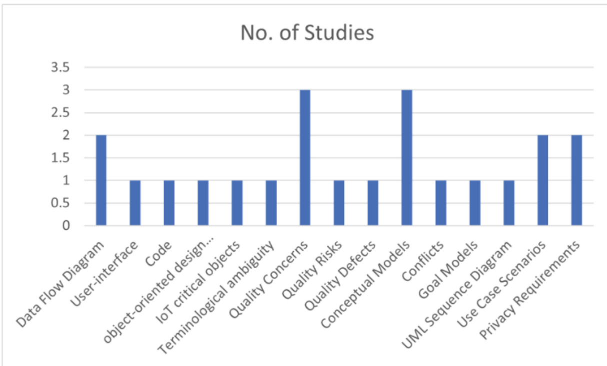


Fig. 1. Information extracted from User Stories

Figure 15: Bar Graph, 2 Dimensions, Single, Figure 1 with caption in "Automated Quality Concerns Extraction from User Stories and Acceptance Criteria for Early Architectural Decisions" [2] page 3 under the heading "Related Work". Visual density = 0.14.

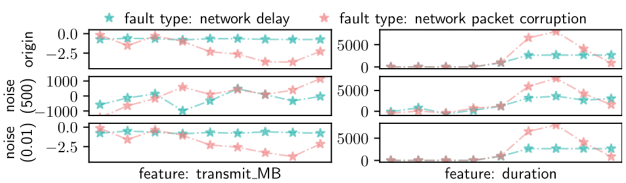
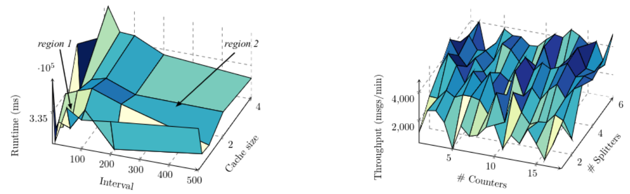


Figure 2: Gaussian noise augmentation at scales of 500 and 0.01. A scale of 500 disrupts the decreasing 'transmit_MB' feature in packet corruption but minimally affects 'duration'.

Figure 16: Line Graph, 2 Dimensions, Multipanel (6 panels), Figure 2 in "The Potential of One-Shot Failure Root Cause Analysis: Collaboration of the Large Language Model and Small Classifier" [27] under the heading "BACKGROUND". Visual density = 0.18



(a) MongoDB (b) Storm

Fig. 1. The projected configuration landscapes of two configurable software systems.

Figure 17: Surface Graphs, 3 Dimensions, Multipanel (2 panels), Figure 1 with caption in "Adapting Multi-objectivized Software Configuration Tuning" [13] under the heading "INTRODUCTION". Visual density = 0.27.

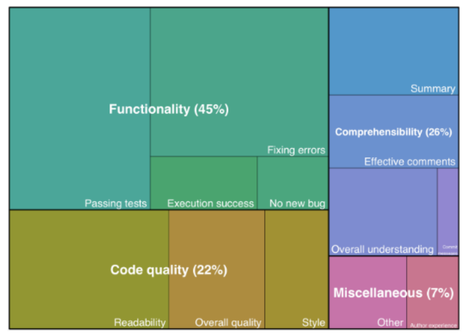


Figure 3: A Breakdown of the top three self-reported factors and their subcategories in code review evaluation, displayed with self-reported frequencies. Three key factors that guide code review decisions are functionality (highlighted by 45% of participants), code quality (26%), and comprehensibility, mainly represented by the quality of the comments and summaries (22%).

Figure 18: Tree Map, 2 Dimensions, Single, Figure 3 with caption in "Exploring the Effects of Urgency and Reputation in Code Review An Eye-Tracking Study" [64] under the heading "DATA ANALYSIS AND RESULTS". Visual density = 0.96.

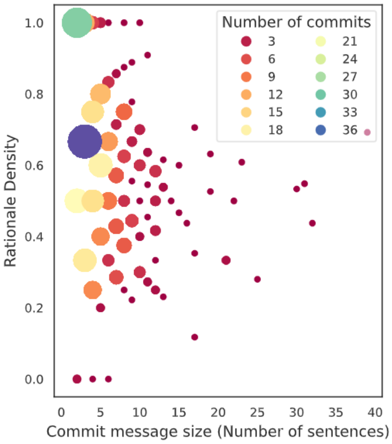


Figure 3: Commit message size versus rationale density

Figure 19: Bubble Chart, 2 Dimensions, Single, Figure 3 with caption in "Rationale Dataset and Analysis for the Commit Messages of the Linux Kernel Out-of-Memory Killer" [21] under the heading "DATASET DESCRIPTION AND ANALYSIS". Visual density = 0.11.

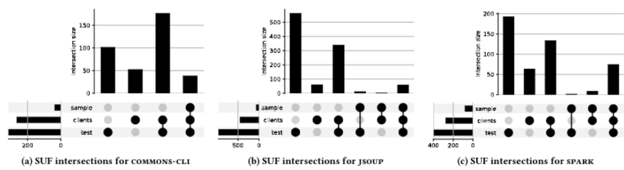


Figure 3: UpSet plots [17] depicting the common and unique uses between third-party clients, tests, and samples for JSOUP, COMMONS-CLI, and SPARK

Figure 20: Dot Array and Bar Chart, 2 Dimensions, Combination, Figure 3 with caption in "Lightweight Syntactic API Usage Analysis with UCov" [43] under the heading "EXPLORATORY CASE STUDY". Visual density = 0.15.

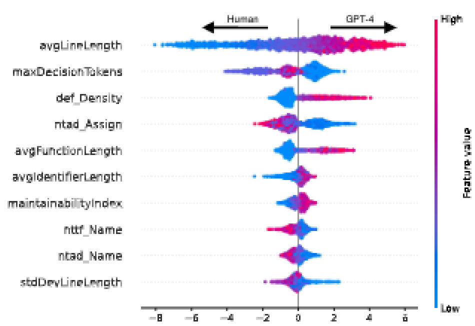


Figure 5: SHAP feature importance of non-gameable features

Figure 21: Dot Plot, 2 Dimensions, Single, Figure 5 with caption in "Whodunit: Classifying Code as Human Authored or GPT-4 generated- A case study on CodeChef problems" [30] under the heading "RESULTS". Visual density = 0.11.

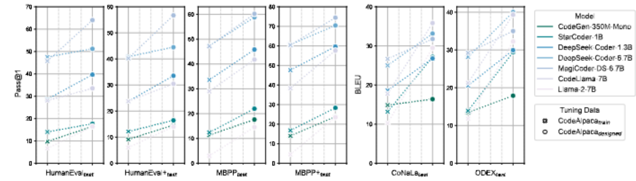
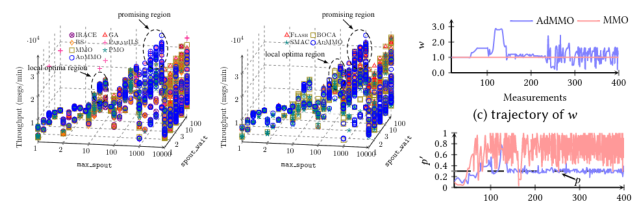


Figure 5: RQ3 - Effectiveness of design alignment. The Y-axis shows the values of the metrics (Pass@1 for the left four plots and BLEU for the right two plots). The X-axis of each subplot represents the datasets before (left-hand side: CodeAlpaca$_{train}$) and after (right-hand side: CodeAlpaca$_{designed}$) applying design alignment. Both datasets are applied to each target model to fine-tune it. Detailed values are available in [1].

Figure 22: Slope Graph, 2 Dimensions, Multipanel (6 panels), Figure 5 with caption in "DataRecipe — How to Cook the Data for CodeLLM?" [36] under the heading "ANSWERING RESEARCH QUESTIONS". Visual density = 0.10.



(a) explored configurations (measurement-based optimizers) (b) explored configurations (model-based optimizers) (c) trajectory of w (d) trajectory of p'

Fig. 7. Example run for system Storm. (a) and (b) are the projected landscape of explored configurations by all optimizers; (c) and (d) are the trajectories of w and the actual proportion of nondominated configurations p', respectively. The dashed line in (d) represents the expected proportion of nondominated configurations.

Figure 23: Scatter Plots, 3 Dimensions, part of Multipanel (4 panels), Figures 7a and 7b with caption in "Adapting Multi-objectivized Software Configuration Tuning" [13] page 17, under the heading "RESULTS AND ANALYSIS". Visual density = 0.20.

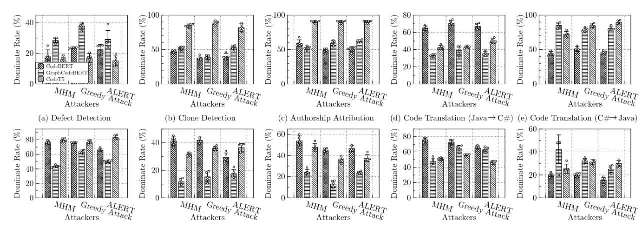


Fig. 7. Dominance rate of MOAA in AE generation across all three objectives over the baseline algorithms.

Figure 24: Grouped Bar Chart, 2 Dimensions, Multipanel (10 panels), Figure 7 with caption in "Evolutionary Multi-objective Optimization for Contextual Adversarial Example Generation" [65] page 13, under the heading "RESULTS AND DISCUSSIONS". Visual density = 0.26.

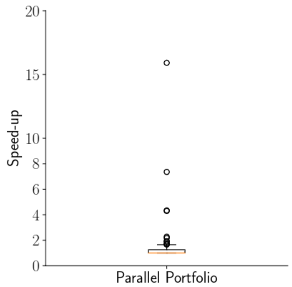


Fig. 10. Speed-up of the parallel portfolio of *DSS* and predicate abstraction compared to standalone predicate abstraction

Figure 25: Box and Whisker-Plot and Scatter Plot, 2 Dimensions, Combination, Figure 10 with caption in "Decomposing Software Verification using Distributed Summary Synthesis" [5] page 16, under the heading "EVALUATION". Visual density = 0.030.

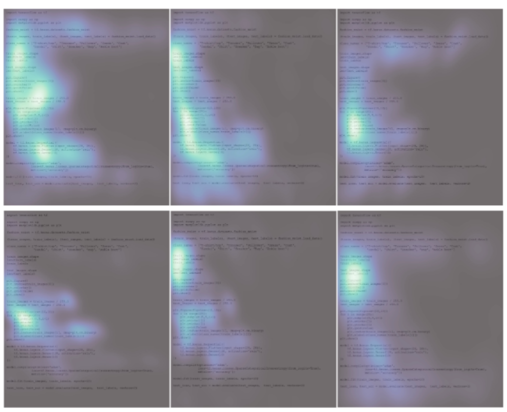


Figure 1: Heatmap of different search and reading strategies. A replay of the gaze tracks showed that the novices (top) showed mostly sequential reading, while the experts (bottom) jumped to selective points of interest.

Figure 26: Heatmap, 2 Dimensions, Multipanel (6 panels), Figure 1 with caption in "An Investigation of How Software Developers Read Machine Learning Code" [61] page 6, under the heading "RESULTS". Visual density = 0.97.

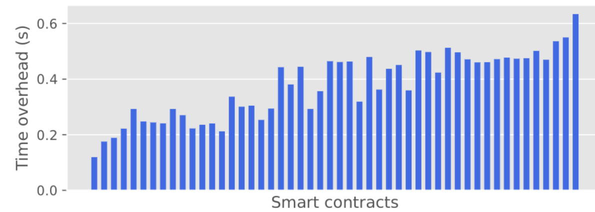


Figure 7: The time overhead of FunRedisp for function redispatch on the 50 smart contracts in the dataset.

Figure 27: Bar Chart, 2 Dimensions, Single, Figure 7 with caption in "FunRedisp: Reordering Function Dispatch in Smart Contract to Reduce Invocation Gas Fees" [41] page 9 under the heading "EVALUATION". Visual density = 0.81.