Same Quality Metrics, Different Graph Drawings

Simon van Wageningen
Utrecht University Department of Information and Computing Sciences, The Netherlands

Tamara Mchedlidze
Utrecht University Department of Information and Computing Sciences, The Netherlands

Alexandru C. Telea
Utrecht University Department of Information and Computing Sciences, The Netherlands

Figure 1 Morphing a graph drawing (START) into six different target shapes (X,VERT,HOR,O,DINO,GRID) while keeping one or more quality metric(s) nearly constant (ST,ELD,CN,AR). Values for ST,ELD,AR are within $\pm\epsilon = 0.0025$ and within $\pm\epsilon = \texttt{CN}(\Gamma) * 0.05$ for CN, so the morphed drawings have very similar quality to the starting ones.

Abstract

Graph drawings are commonly used to visualize relational data. User understanding and performance are linked to the quality of such drawings, which is measured by quality metrics. The tacit knowledge in the graph drawing community about these quality metrics is that they are not always able to *accurately* capture the quality of graph drawings. In particular, such metrics may rate drawings with very poor quality as very good. In this work we make this tacit knowledge explicit by showing that we can modify existing graph drawings into arbitrary target shapes while keeping one or more quality metrics almost identical. This supports the claim that more advanced quality metrics are needed to capture the 'goodness' of a graph drawing and that we cannot confidently rely on the value of a single (or several) certain quality metrics.

2012 ACM Subject Classification Human-centered computing → Graph drawings

Keywords and phrases graph drawing, quality metrics, assumptions, fooling

Digital Object Identifier 10.4230/LIPIcs.GD.2025.5

Related Version *Full Version*: https://arxiv.org/abs/2508.15557

Supplementary Material *Software (Source Code)*: https://github.com/simonvw95/Same-Quality-Metric--Different-Graph-Drawings

1 Introduction

Relational data is typically visualized using node-link diagrams, commonly known as *graph drawings* (GD). High-quality graph drawings are key to user understanding and performance [31]. In turn, such quality is measured by various so-called *quality metrics* (QM) [30]. At

© Simon van Wageningen, Tamara Mchedlidze, and Alexandru C. Telea;
licensed under Creative Commons License CC-BY 4.0
33rd International Symposium on Graph Drawing and Network Visualization (GD 2025).
Editors: Vida Dujmović and Fabrizio Montecchiani; Article No. 5; pp. 5:1–5:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

a high level QMs capture two different aspects of the goodness of a GD: 1. how readable the elements of the drawing are, e.g., few crossings, few node overlaps – we call these *readability* metrics; and 2. how well the graph structure is represented by the drawing – *faithfulness metrics*. For instance, nodes lie at distances similar to their graph theoretical distances (measured by *stress*) or clusters are clearly depicted [28]

Quality metrics tend to correlate – often by design – with the aesthetic preferences of users [18, 31] and task performance [29]. As such, GD algorithms aim to (in)directly optimize them in order to produce pleasing drawings. For instance, spring-based techniques like Fruchterman-Reingold [15] and ForceAtlas2 [20] use forces on their nodes (and edges) in various ways, leading to drawings with 'good' scores on most GD quality metrics. Other techniques like Stress Majorization [16], Stress-Plus-X [11], DeepGD [36], SGD^2 [1], and Core-GD [17] directly optimize one or more metrics to produce aesthetically pleasing drawings.

Two underlying and often implicit assumptions on quality metrics are that a 'high' value indicates a 'good' drawing; and that drawings of the same graph with similar metric values have similar quality. It is, however, widely known by Graph Drawing researchers, and has been exemplified multiple times by the Graph Drawing live challenge [12], that these assumptions do not always hold. In particular, over the years, GD challenge participants have presented drawings with very high QMs values that had an obfuscating appearance and did not reveal the graph structure.

We devote this work to make this tacit knowledge explicit and show that we cannot confidently rely on the value of a single quality metric nor on combinations of certain quality metrics. For this, we *'fool'* the quality metrics by morphing given, high-quality graph drawings into a given target shape (which can be any arbitrary shape) using a simulated annealing process while keeping one or more given quality metric values relatively fixed. Thus we are able to obtain drawings that obfuscate the graph structure and have very high values of quality metrics. While for readability metrics this is rather a confirmation of the tacit knowledge we already possess, it is very surprising to achieve this for a faithfulness quality metric such as stress as well as for combinations of certain QMs. We also show the limitations of our approach – certain combinations of metrics can not be fooled by it. Finally, we experiment with graphs of varying structures to illustrate how the graph's structure affects the difficulty of fooling quality metrics.

2 Related Work

The assumption that quality metrics directly correlate with human understanding, perception, and preference has been researched by many previous (user) studies [4]. The authors in [31] showed that larger numbers of edge crossings and edge bends are negatively correlated with human perception. Huang [18] refined such findings by using eye tracking to show that small edge angles lead to poorer performance in path search tasks.

The number of edge crossings in a drawing has been shown to correlate with user preferences [9, 31]. The perception of how well nodes are placed according to their shortest path, also called *stress*, was researched in a 2024 study [26], which found that users are able to perceive different values of stress in different drawings. Furthermore, stress also tends to correlate with user preferences [9].

Several types of visualizations beyond GD have been put to the test to try and fool statistical properties. Many works [2, 3, 7, 24] have shown that it is possible to control the visual appearance of 2D scatterplots, by iteratively morphing these into target shapes, while keeping summary statistical metrics such as means, standard deviations and correlation

similar. Different DR algorithms can create wildly different projections of the same dataset which have very similar quality metric values [14]. Separately, Machado, Telea, and Behrisch showed that one can control the shapes of clusters in projections and only marginally affect quality metrics [23], as well as completely distort the visual appearance of projections without affecting multiple quality metrics [22].

Similarly, Chari and Pachter [6] showed that they could transform projections of the same dataset into any specified shape without heavily adjusting the quality metric values. While such work did not aim to technically 'fool' quality metrics, it provides strong evidence that metrics and projection appearance are not inherently correlated – an effect that we also show, in our work, to hold for graph drawings.

Regarding graphs (but not their drawings), Chen et al. [8] showed that one can generate graphs with the exact same graph statistics, such as the number of nodes, edges, and triangles. Di Bartolomeo, Lang and Dunne [13] (potentially a satirical paper) suggested as future research directions the idea of 'rocketshipness', in which a graph drawing is optimized to resemble a rocket ship while maintaining one or more GD quality metrics. However, this idea was not further explored.

3 Fooling GD metrics: Experimental setup

3.1 Preliminaries

We first introduce a few notations and concepts. An *undirected graph* $G = (V, E)$ is a set of nodes $V = \{v_1, \ldots, v_n\}$ and a set of undirected unweighted edges $E = \{e_1, \ldots, e_m\} \subseteq V \times V$. A straight-line *graph drawing* Γ of G assigns 2-dimensional coordinates X_i to nodes $v_i \in V$ and line segments L_i to edges $e_i \in E$, so Γ can be represented as a matrix $X^2 \in \mathbb{R}^{n\times 2}$ with rows $X_i \in \mathbb{R}^2$. Let $D \in \mathbb{R}^{n\times n}$ denote the *shortest path matrix* of graph-theoretic distances d_{ij} between all node-pairs $(v_i, v_j) \in V \times V$. We next refer to the Euclidean distance $\|X_i - X_j\|$ between two nodes v_i and v_j simply as *distance*. Let $deg(v)$ be the degree of node v, *i.e.*, the number of edges incident to v. A *quality metric* is a function $Q(\Gamma) \in [0, 1]$ which assigns a value to Γ, with low (resp. high) values denoting better (resp. poorer) drawings.

3.2 Metrics

We consider in our work the following GD quality metrics and their combinations:

Stress: Measures the difference between the Euclidean distances of all node-pairs and their shortest path distances [21] as $\texttt{ST}(\Gamma) = \frac{1}{n(n-1)/2}\sum_{i=0}^{n}\sum_{j=i+1}^{n}\frac{(\|Z_i - Z_j\| - d_{ij})^2}{d_{ij}^2}$, where Z_i scales the coordinate X_i by the shortest path distances as $Z_i = \frac{\sum_{i\neq j}\|X_i - X_j\|/d_{ij}}{\sum_{i\neq j}\|X_i - X_j\|^2/d_{ij}^2}X_i$.

$\texttt{ST}(\Gamma) = \frac{1}{n(n-1)/2}\sum_{i=0}^{n}\sum_{j=0}^{i}\frac{(\|Z_i - Z_j\| - d_{ij})^2}{d_{ij}^2}$,

We expect the ST metric to be the most difficult metric to fool in drawings, as it explicitly evaluates how well graph structure is depicted in a drawing, i.e., is a faithfulness QM.

Edge length deviation: Measures the average deviation of edge lengths [25] from the mean μ of all lengths in a drawing as $\texttt{ELD}(\Gamma) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\|L_i\| - \mu)^2}$. We expect this metric to be relatively easy to fool as it *solely* focuses on the uniformity of edge lengths.

Crossing number: We capture this by the number of intersections of all edge segments L_i, *i.e.* $\texttt{CN}(\Gamma) = \sum_{i=0}^{m} \sum_{j=i+1}^{m} \mathbf{1}(L_i \cap L_j \neq \emptyset)$. We expect that CN is hard to fool, as making subtle changes to any drawing tends to have a strong impact on how its lines cross.

Angular resolution: We measure the frequency of small angles occurrences in a graph drawing by computing the average deviation of angles [25] of adjacent edges *vs* the best possible angle as $\texttt{AR} = \frac{1}{n'} \sum_{i=1}^{n'} \left| \frac{\Theta_i - \theta_i}{\Theta_i} \right|$, where n' is the number of nodes with degree ≥ 2; θ_i is the smallest measured angle between consecutive adjacent edges of node i and $\Theta_i = 2\pi/deg(v_i)$ is the best possible angle between consecutive edges for node i. We speculate that this metric is easy to fool as it is easy to keep similar angles while making large visual changes.

3.3 Simulated Annealing Approach

We follow the approach in [24] to slowly morph any given graph drawing Γ with node coordinates X into a drawing Γ' with node coordinates X', while keeping one or more quality metrics values $QM^i(\Gamma')$, $i \in \{a, b, \dots\}$ very close to their original values $QM^i(\Gamma)$ and moving X' closer to a target shape Y. Here, Y can be *any* set of node coordinates, as long as $|Y| = |X|$, as this simplifies our morphing implementation.

Algorithm 1 Simulated Annealing Morphing

1: $DIFF_{curr} \leftarrow Sim(X, Y)$
2: $\mathbf{qm} \leftarrow [QM^a(\Gamma), QM^b(\Gamma), \dots]$
3: **for** $i = 1$ to N_{max} **do**
4: $\quad (X', DIFF_{test}) \leftarrow Jit(X, T_i, DIFF_{curr}, Y)$
5: $\quad \mathbf{qm}' \leftarrow [QM^a(\Gamma'), QM^b(\Gamma'), \dots]$
6: $\quad$ **if** $\forall j,\ |\mathbf{qm}'[j] - \mathbf{qm}[j]| \leq \epsilon_{\mathbf{j}}$ **then**
7: $\quad\quad X \leftarrow X'$
8: $\quad\quad DIFF_{curr} \leftarrow DIFF_{test}$
9: $\quad$ **end if**
10: **end for**

Algorithm 1 details our morphing: We start by computing the similarity of our graph drawing to the target shape and its quality metric value (lines 1-2). The *Jit* function jitters a random selection of nodes $X^s \subset X, 1 \leq |X|/15$ by a random value drawn from a normal distribution $[-0.5, 0.5]/25$ while increasing the similarity of X' to Y (line 4, see also Alg. 2). We accept X' if the quality $\mathbf{qm}'$ of its drawing Γ' is within a small range ϵ of the initial quality $\mathbf{qm}$ (lines 5-8). We repeat this process for $N_{max} = 30000$ iterations. To escape local minima, we also use simulated annealing: If a randomly generated value is below the current temperature T_i, we accept the node coordinates X_{jit}, irrespective of its (dis)similarity to Y. T_i varies over iterations: We start with $T_{init} = 0.4$ and quadratically decrease it to a final value $T_{N_{max}} = 0.001$. After exploring the practical ranges of the metrics in [25] and their variations discussed in [33], we set $\epsilon = 0.0025$ for ST,ELD,AR and $\epsilon = \texttt{CN}(\Gamma) * 0.05$ for CN.

For the similarity $Sim(X, Y)$ (see Alg. 3), one can use Mean Squared Error $MSE = \frac{1}{n} \sum_{i=1}^{n} (X_i - Y_i)^2$ or the Procrustes Statistic [37] (similar to *MSE* but invariant to rotation, translation and scaling). We also briefly experimented with the *Wasserstein Distance* [35] and the *Sinkhorn* [10] algorithm. However, these approaches yielded slightly worse results and were thus discarded.

Algorithm 2 Jitter function $Jit(X, T, DIFF, Y)$

```
while do
    X^s ⊂ X, 1 ≤ |X|/15
    X_jit ← X + (X^s + rand(N(−0.5,0.5))/25)
    if Sim(X_jit, Y) < DIFF ∨ T > rand(U(0,1)) then
        return (X_jit, Sim(X_jit, Y))
    end if
end while
```

The authors of [24] used the average distance of all points in X to the closest point in Y, which gives more morphing freedom. Our Sim design slightly adapts this idea. Algorithm 3 details our Similarity function: We start by computing the closest distances from two sets of node coordinates X and Y. We compute the distance of node X_0 (the *current* first node) in coordinate set X to all nodes in coordinate set Y. The distance to the closest node Y_{idx} in Y is then added to the total loss after which both X_0 and Y_{idx} are removed from X and Y, respectively. This iterative process is then repeated until all nodes are removed from the coordinate set and ensures no repetition of nodes in any pairs. To compare similarity values across different graphs, we express the similarity in a percentage, where we consider perfect similarity to the target shape ($Sim(Y, Y) = 0$) as 100% and perfect similarity to the starting shape as 0%, i.e.: $100 - (Sim(X', Y)/Sim(X, Y) * 100)$ where $Sim(X, Y) \rightarrow \mathbb{R}$ is equal to the starting similarity from Alg 1 (line 1).

Algorithm 3 Similarity function $Sim(X, Y)$

```
loss ← 0
while n > 0 do
    DIS ← [‖X_0 − Y‖]_{i=0}^n
    (idx, dis) ← (argmin(DIS), min(DIS))
    X ← RemoveRow(X, 0)
    Y ← RemoveRow(Y, idx)
    loss ← loss + dis
    n ← n − 1
end while
return loss
```

3.4 Datasets

We perform experiments on the five graphs in Tab. 1. Four come from the Network Repository [32] due to their interesting varying structures. We created the fifth one using the

Graph	n	m	$deg(V)_{avg}$	*source*
polbooks	105	441	8.40	[32]
lnsp_131	123	275	4.47	[32]
bar-albert	142	175	2.46	generated
gams10am	171	298	3.49	[32]
dwt_307	307	1108	7.22	[32]

Table 1 Descriptive statistics of graphs used in our experiments.

dual-Barabasi-Albert algorithm [27] to control the node count n. For each graph we create initial drawings (START) using ForceAtlas2 [20] with 5000 iterations and default parameters. We translate and scale all drawings to have coordinates in $[0, 1]$ so that *Jit* becomes scale invariant. We use six target shapes Y from [24] including the Dinosaur shape from [5], see Fig. 2. For all shapes we create simple algorithms that generate Y for any point count n so we can next easily enforce $|X| = |Y|$ (see Sec. 3.3). Our data and code are available on GitHub [34].

Figure 2 Target shapes used in our experiments.

4 Results and Discussion

We next refer to results from the figures and tables in the Appendix with the ⋆ notation. For larger versions of these figures check the full paper version. These extra results include the drawings of all pairs and triples of metric combinations, distribution of similarity percentages, and results of statistical testing. We also refer the reader to our video material, in which we replicate a dancing person while keeping a quality metric value the same. An example of four frames of this video can be seen in Fig. 3. Additional videos and details on how the videos were created can be found on our GitHub page [34].

Individual metrics: Figures 1 and 4 show the resulting drawings of fooling individual metrics and all four metrics at once. We see that not all individual quality metrics can be fooled equally well. As speculated, fooling ST proves to be the most difficult. For example, the *bar-albert* graph nearly perfectly morphs to all six targets for ELD,CN,AR but not for ST. We see similar results for *polbooks*, *lnsp_131* and *dwt_307*. For the metrics ELD and AR, we observe easy fooling for all graphs except *dwt_307*. We also see that fooling the number of crossings (CN) is easier than expected – see the results for all graphs except *gams01am* and *dwt_307*. Aggregating the results over all graphs and target shapes (see Fig. 7⋆) shows that ELD is much easier to fool than ST; and AR is much easier to fool than ST and CN.

Figure 3 Four sequential target frames from the video and the resulting sequential drawings from attempting to fool a quality metric (ELD) for the target frames.

Figure 4 Collection of different graph drawings. The START column indicates the starting drawing of various graphs. The results of applying Alg. 1 to START to six target shapes (X, VERT, HOR, O, DINO, GRID) are shown in their respective columns. The rows indicate the (combinations of) metrics that have $\pm\epsilon = 0.0025$ for ST,ELD,AR and $\pm\epsilon = \texttt{CN}(\Gamma) * 0.05$ for CN.

Combination of metrics: We observe that our technique is almost never able to fool all four metrics at once, except for the *bar-albert* graph. Looking at the drawings in Fig. 5⋆ and

the similarity values in Fig. 6⋆, we see that fooling three metrics at the same time is also hard. In contrast, fooling combinations of two metrics is sometimes possible but proves to be hard for *dwt_307* and *gam10am*. We observe that the combination `ST-CN` is rather resistant to fooling; every other metric pair is much easier to fool than `ST-CN` except for `ELD-CN`.

Graph type: The spread of the similarity values in Fig. 6⋆, and in particular the low values for *dwt_307* and *bar-albert*, support our hypothesis that the graph *type* can influence how difficult it is to fool its quality metrics. For *dwt_307*, we see that the target shapes (except `GRID`) are much less visible for most metrics compared to results of other graphs. In Fig. 8⋆, we see that the metrics and their combinations for this graph are significantly harder to fool than other graphs, most likely due to its strong mesh-like structure. The metrics and their combinations for the *bar-albert* graph, whose tree-like structure allows for a lot of layout flexibility, are significantly easier to fool compared to all other graphs in our dataset.

Target shape: Lastly, we observe that some target *shapes* are harder to morph to. For instance, `DINO` is much more complex than `O` or `X`. This leads to messy drawings for some graphs – *dwt_307* is unable to even come close. When aggregating the results over all graphs and metric combinations (see Fig. 8⋆), we see that only the `GRID` shape is significantly easier to morph to than all other shapes.

5 Conclusion

In this work we provide evidence against the assumptions that (1) 'high' quality metric values indicate 'good' graph drawings; and that (2) drawings of the same graph with similar metric values are similar in their quality. We use simulated annealing to morph existing graph drawings into six different, arbitrary, target shapes without substantially altering one or more quality metric values. To our knowledge, this is the first time that a systematic fooling of graph drawing quality metrics has been explored. Depending on the graph and metric, most graph drawings can easily be morphed towards such targets. We observe that the structure of the graph has an influence on how well a graph drawing can fooled. Furthermore, we find evidence that some metrics (*stress*) are more difficult to fool than others, depending on the target shape; and that some combinations of metrics are easier to fool. All in all, our results suggest that current quality metrics are not enough to capture the essence of a 'good' graph drawing on their own. Similar to Huang et al. [19], we emphasize the importance of combinations of two or more quality metrics that capture both the graph structure (faithfulness) and readability of the drawing. Future work can consider advancing our fooling method by using more sophisticated techniques such as gradient descent or deep learning approaches. Furthermore, more experiments can be conducted with more graphs, target drawings, and quality metrics, as well as linear combinations of multiple quality metrics. Lastly, a perceptual user-study could explore how readable or misleading these morphed drawings are.

References

1 Reyan Ahmed, Felice De Luca, Sabin Devkota, Stephen Kobourov, and Mingwei Li. Multicriteria Scalable Graph Drawing via Stochastic Gradient Descent, $(SGD)^2$. *IEEE TVCG*, 28(06):2388–2399, 2022. doi:10.1109/TVCG.2022.3155564.

2 Francis J. Anscombe. Graphs in Statistical Analysis. *The American Statistician*, 27(1):17–21, 1973. doi:10.1080/00031305.1973.10478966.

3 Tal Boger, Steven B. Most, and Steven L. Franconeri. Jurassic Mark: Inattentional Blindness for a Datasaurus Reveals that Visualizations are Explored, not Seen. In *IEEE Vis*, pages 71–75, 2021. doi:10.1109/VIS49827.2021.9623273.

4 Michael Burch, Weidong Huang, Mathew Wakefield, Helen C. Purchase, Daniel Weiskopf, and Jie Hua. The State of the Art in Empirical User Evaluation of Graph Visualizations. *IEEE Access*, 9:4173–4198, 2021. doi:10.1109/ACCESS.2020.3047616.

5 Albert Cairo. Download the Datasaurus: Never trust summary statistics alone; always visualize your data, 2016. URL: https://thefunctionalart.blogspot.com/2016/08/download-datasaurus-never-trust-summary.html.

6 Tara Chari and Lior Pachter. The specious art of single-cell genomics. *Computational Biology*, 19(8):1–20, 2023. doi:10.1371/journal.pcbi.1011288.

7 Sangit Chatterjee and Aykut Firat. Generating Data with Identical Statistics but Dissimilar Graphics. *The American Statistician*, 61(3):248–254, 2007. doi:10.1198/000313007X220057.

8 Hang Chen, Utkarsh Soni, Yafeng Lu, Vahan Huroyan, Ross Maciejewski, and Stephen G. Kobourov. Same Stats, Different Graphs: Exploring the Space of Graphs in Terms of Graph Properties . *IEEE TVCG*, 27(03):2056–2072, 2021. doi:10.1109/TVCG.2019.2946558.

9 Markus Chimani, Patrick Eades, Peter Eades, Seok-Hee Hong, Weidong Huang, Karsten Klein, Michael R. Marner, Ross T. Smith, and Bruce H. Thomas. People prefer less stress and fewer crossings. In *Graph Drawing*, volume 8871 of *LNCS*, pages 523–524. Springer Berlin Heidelberg, 2014.

10 Marco Cuturi. Sinkhorn distances: lightspeed computation of optimal transport. In *NIPS*, volume 2, page 2292–2300. Curran Associates Inc., 2013. doi:10.48550/arXiv.1306.0895.

11 Sabin Devkota, Reyan Ahmed, Felice De Luca, Katherine E. Isaacs, and Stephen G. Kobourov. Stress-Plus-X (SPX) Graph Layout. In *Graph Drawing*, volume 11904 of *LNCS*, pages 291–304. Springer Berlin Heidelberg, 2019. doi:10.1007/978-3-030-35802-0_23.

12 Sara Di Bartolomeo, Fabian Klute, Debajyoti Mondal, and Jules Wulms. Graph drawing contest report. In *Graph Drawing*, LIPIcs, pages 41:1–41:13. Schloss Dagstuhl, October 2024. doi:10.4230/LIPICS.GD.2024.41.

13 Sara Di Bartolomeo, Matěj Lang, and Cody Dunne. The worst graph layout algorithm ever. In *Proc. alt. VIS workshop at IEEE VIS*, 2022. doi:10.31219/osf.io/4hfy9.

14 Mateus Espadoto, Rafael M. Martins, Andreas Kerren, Nina S. T. Hirata, and Alexandru C. Telea. Toward a Quantitative Survey of Dimension Reduction Techniques. *IEEE TVCG*, 27(3):2153–2173, 2919. doi:10.1109/TVCG.2019.2944182.

15 Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991. doi:10.1002/spe.4380211102.

16 Emden R. Gansner, Yehuda Koren, and Stephen C. North. Graph Drawing by Stress Majorization. In *Graph Drawing*, volume 3383 of *LNCS*, page 239–250. Springer Berlin Heidelberg, 2005. doi:10.1007/978-3-540-31843-9_25.

17 Florian Grötschla, Joël Mathys, Robert Veres, and Roger Wattenhofer. Core-GD: A hierarchical framework for scalable graph visualization with GNNs. In *The Twelfth International Conference on Learning Representations*, 2024. doi:10.48550/arXiv.2402.06706.

18 Weidong Huang. Using eye tracking to investigate graph layout effects. In *APVIS*, pages 97–100. IEEE Computer Society, 2007. doi:10.1109/APVIS.2007.329282.

19 Weidong Huang, Peter Eades, Seok-Hee Hong, and Chun-Cheng Lin. Improving multiple aesthetics produces better graph drawings. *Journal of Visual Languages and Computing*, 24(4):262–272, 2013. doi:10.1016/j.jvlc.2011.12.002.

20 Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software. *PLOS ONE*, 9(6):e98679, 2014. doi:10.1371/journal.pone.0098679.

21 Tomihisa Kamada and Satoru Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989. doi:10.1016/0020-0190(89)90102-6.

22 Alister Machado, Michael Behrisch, and Alexandru C. Telea. Necessary but not sufficient: Limitations of projection quality metrics. *Computer Graphics Forum*, page e70101, 2015. doi:10.1111/cgf.70101.

23 Alister Machado, Alexandru C. Telea, and Michael Behrisch. ShaRP: Shape-Regularized Multidimensional Projections. In *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2023. doi:10.2312/eurova.20231088.

24 Justin Matejka and George Fitzmaurice. Same Stats, Different Graphs: Generating Datasets with Varied Appearance and Identical Statistics through Simulated Annealing. In *CHI*, page 1290–1294. Association for Computing Machinery, 2017. doi:10.1145/3025453.3025912.

25 Gavin J. Mooney, Helen C. Purchase, Michael Wybrow, and Stephen G. Kobourov. The Multi-Dimensional Landscape of Graph Drawing Metrics. In *IEEE PacificVis*, pages 122–131, 2024. doi:10.1109/PacificVis60374.2024.00022.

26 Gavin J. Mooney, Helen C. Purchase, Michael Wybrow, Stephen G. Kobourov, and Jacob Miller. The Perception of Stress in Graph Drawings. In *Graph Drawing*, volume 320 of *LIPIcs*, pages 21:1–21:17. Schloss Dagstuhl, 2024. doi:10.4230/LIPIcs.GD.2024.21.

27 Niema Moshiri. The dual-Barabási-Albert model. *arXiv preprint arXiv:1810.10538*, 2018.

28 Quan Nguyen, Peter Eades, and Seok-Hee Hong. On the faithfulness of graph visualizations. In *2013 IEEE PacificVis*, pages 209–216, 2013. doi:10.1109/PacificVis.2013.6596147.

29 Helen C. Purchase. Which aesthetic has the greatest effect on human understanding? In *Graph Drawing*, volume 1353 of *LNCS*, page 248–261. Springer Berlin Heidelberg, 1997. doi:10.1007/3-540-63938-1_67.

30 Helen C. Purchase. Metrics for Graph Drawing Aesthetics. *Journal of Visual Languages and Computing*, 13(5):501–516, 2002. doi:10.1006/jvlc.2002.0232.

31 Helen C. Purchase, Robert F. Cohen, and Murray James. Validating graph drawing aesthetics. In *Graph Drawing*, volume 1027 of *LNCS*, pages 435–446. Springer Berlin Heidelberg, 1996. doi:10.1007/bfb0021827.

32 Ryan A. Rossi and Nesreen K. Ahmed. The Network Data Repository with Interactive Graph Analytics and Visualization. In *AAAI*, 2015. URL: https://networkrepository.com.

33 Simon van Wageningen, Tamara Mchedlidze, and Alexandru C. Telea. An Experimental Evaluation of Viewpoint-Based 3D Graph Drawing. *Computer Graphics Forum*, 43(3):e15077, 2024. doi:10.1111/cgf.15077.

34 Simon van Wageningen, Tamara Mchedlidze, and Alexandru C. Telea. Same Quality Metrics source code and videos, 2025. URL: https://github.com/simonvw95/Same-Quality-Metric--Different-Graph-Drawings.

35 Cédric Villani. *The Wasserstein distances*, pages 93–111. Springer, 2009. doi:10.1007/978-3-540-71050-9_6.

36 Xiaoqi Wang, Kevin Yen, Yifan Hu, and Han-Wei Shen. DeepGD: A deep learning framework for graph drawing using GNN. *IEEE CGA*, 41(05):32–44, 2021. doi:10.1109/MCG.2021.3093908.

37 Yong Wang, Zhihua Jin, Qianwen Wang, Weiwei Cui, Tengfei Ma, and Huamin Qu. DeepDrawing: A Deep Learning Approach to Graph Drawing. *IEEE TVCG*, pages 1–1, 2019. doi:10.1109/TVCG.2019.2934798.

6 Appendix

6.1 Combinations of Metrics

Figure 5 Collection of different graph drawings. The START column indicates the starting drawing of various graphs. The results of applying Alg. 1 to START to six target shapes (X, VERT, HOR, O, DINO, GRID) are shown in their respective columns. The rows indicate the metrics that have $\pm\epsilon = 0.0025$ for combinations of ST,ELD,AR and $\pm\epsilon = \mathtt{CN}(\Gamma) * 0.05$ for CN.

6.2 Similarity percentages

Figure 6⋆ visualizes the distributions of the similarity percentages. Here, each dot represents how similar the fooled drawing is to its target shape.

Figure 6 Distributions of how similar the resulting drawings are to the six target shapes.

Figure 7 Comparing the distributions of each metric's results with each other metric. Each box indicates that the resulting distribution of the results of the row metric minus the results of the column metric are non-significantly (gray) and significantly (yellow) greater (easier to fool) than a distribution symmetric about zero.

6.2.1 Metric comparison

To test significant differences between the non-normally distributed results of the fooled metrics we first employ the Friedman test. We find that there are significant differences ($\chi^2 = 314$, $p < 0.05$) between the results of metrics. Since our data is paired, we then perform pairwise metric comparisons using the Wilcoxon signed-rank test. Additionally, we correct the p-values using the Bonferroni adjustments.

In Figure 7⋆ we display the results of the significance tests with $p < 0.05$. Here, the metric of a row is compared with the metrics of all columns and vice-versa. A yellow (resp., gray) color indicates that the resulting distribution of the row metric minus the column metric is significantly (resp., or non-significantly) greater than a distribution symmetric about zero. As we are using a *greater than* alternative hypothesis, the pattern in Figure 7⋆ is not symmetric. For example, for `ELD` and `ST` we see that there is a significant difference, meaning that the difference between the distributions of `ELD` and `ST` is greater than a distribution around zero, i.e. `ELD` is easier to fool than `ST`. However, that means that the difference between the distributions of `ST` and `ELD` is smaller than a distribution around zero which is why this pair is not significant for the "greater than" test.

From this pair-wise metric comparison, we observe that fooling individual metrics is easier than fooling combinations, with `ELD` and `AR` being the simplest. For pairs of quality metrics, the `ST-CN` combination is more resistant to fooling than others. We see the same for any other larger combination that includes `ST` and `CN`, such as `ST-ELD-CN` and `ST-CN-AR` which are the most difficult to fool.

6.2.2 Target shape comparison

We test the differences between graphs in exactly the same manner as for the metric comparison, with now the dependent variable being the target shape. The Friedman test shows that there are significant differences between the results of the target shapes ($\chi^2 = 42.9$, $p < 0.05$). The results of the pairwise Wilcoxon signed-rank tests are visualized in Figure 8⋆. From these results we observe that the `GRID` shape is significantly easier to morph into than any other shape. All the other shapes are more or less equally difficult to morph any drawing into, with the exception that the `DINO` shape is significantly easier than `VERT`.

Figure 8 Comparing the distributions of each target's results with each other target (left) and comparing each graph with each other graph (right). Each box indicates that the resulting distribution of the results of the row metric target the results of the column target are non-significantly (gray) and significantly (yellow) greater (easier to fool) than a distribution symmetric about zero.

6.2.3 Graph comparison

We test the differences between graphs in a similar way to the metric comparison, with now the dependent variable being the graph. However, due to the graph data being independent samples we use the Mann-Whitney U test to determine significant differences between pairs of graphs. These results are visualized in Figure 8⋆. From these results we observe that for the *bar-albert* graph it significantly easier to fool any metric combination than it is for any other graph. Conversely, fooling metrics for the *dwt_307* graph is significantly more difficult than for any other graph