



Aesthetic Discrimination of Graph Layouts

Moritz Klammler¹, Tamara Mchedlidze^{1(✉)}, and Alexey Pak²

¹ Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany
moritz@klammler.eu, mched@iti.uka.de

² Fraunhofer Institute of Optronics, System Technologies and Image Exploitation,
Fraunhoferstraße 1, 76131 Karlsruhe, Germany
alexey.pak@iosb.fraunhofer.de

Abstract. This paper addresses the following basic question: given two layouts of the same graph, which one is more aesthetically pleasing? We propose a neural network-based discriminator model trained on a labeled dataset that decides which of two layouts has a higher aesthetic quality. The feature vectors used as inputs to the model are based on known graph drawing quality metrics, classical statistics, information-theoretical quantities, and two-point statistics inspired by methods of condensed matter physics. The large corpus of layout pairs used for training and testing is constructed using force-directed drawing algorithms and the layouts that naturally stem from the process of graph generation. It is further extended using data augmentation techniques. Our model demonstrates a mean prediction accuracy of 96.48%, outperforming discriminators based on stress and on the linear combination of popular quality metrics by a small but statistically significant margin.

The full version of the paper including the appendix with additional illustrations is available at <https://arxiv.org/abs/1809.01017>.

Keywords: Graph drawing · Graph drawing aesthetics
Machine learning · Neural networks · Graph drawing syndromes

1 Introduction

What makes a drawing of a graph aesthetically pleasing? This admittedly vague question is central to the field of Graph Drawing which has over its history suggested numerous answers. Borrowing ideas from Mathematics, Physics, Arts, etc., many researchers have tried to formalize the elusive concept of aesthetics.

In particular, dozens of formulas collectively known as *drawing aesthetics* (or, more precisely, *quality metrics* [6]) have been proposed that attempt to capture in a single number how beautiful, readable and clear a drawing of an abstract graph is. Of those, simple metrics such as the number of edge crossings, minimum crossing angle, vertex distribution or angular resolution parameters, are obviously incapable *per se* of providing the ultimate aesthetic statement.

Advanced metrics may represent, for example, the energy of a corresponding system of physical bodies [5, 9]. This approach underlies many popular graph drawing algorithms [39] and often leads to pleasing results in practice. However, it is known that low values of energy or stress do not always correspond to the highest degree of symmetry [43] which is an important aesthetic criterion [30].

Another direction of research aims to narrow the scope of the original question to specific application domains, focusing on the purpose of a drawing or possible user actions it may facilitate (*tasks*). The target parameters – readability and the clarity of representation – may be assessed via user performance studies. However, even in this case such aesthetic notions as symmetry still remain important [30]. In general, aesthetically pleasing designs are known to positively affect the apparent and the actual usability [25, 41] of interfaces and induce positive mental states of users, enhancing their problem-solving abilities [8].

In this work, we offer an alternative perspective on the aesthetics of graph drawings. First, we address a slightly modified question: “Of two given drawings of the same graph, which one is more aesthetically pleasing?”. With that, we implicitly admit that “the ultimate” quality metric may not exist and one can hope for at most a (partial) ordering. Instead of a metric, we therefore search for a binary *discriminator function* of graph drawings. As limited as it is, it could be useful for practical applications such as picking the best answer out of outputs of several drawing algorithms or resolving local minima in layout optimization.

Second, like Huang et al. [13], we believe that by combining multiple metrics computed for each drawing, one has a better chance of capturing complex aesthetic properties. We thus also consider a “meta-algorithm” that aggregates several “input” metrics into a single value. However, unlike the recipe by Huang et al., we do not specify the form of this combination *a priori* but let an artificial neural network “learn” it based on a sample of labeled training data. In the recent years, machine learning techniques have proven useful in such aesthetics-related tasks as assessing the appeal of 3D shapes [4] or cropping photos [24]. Our network architecture is based on a so-called *Siamese neural network* [3] – a generic model specifically designed for binary functions of same-kind inputs.

Finally, we acknowledge that any simple or complex input metric may become crucial to the answer in some cases that are hard to predict *a priori*. We therefore implement as many input metrics as we can and relegate their ranking to the model. In addition to those known from the literature, we implement a few novel metrics inspired by statistical tools used in Condensed Matter Physics and Crystallography, which we expect to be helpful in capturing the symmetry, balance, and salient structures in large graphs. These metrics are based on so-called *syndromes* – variable-size multi-sets of numbers computed for a graph or its drawing (e.g. vertex coordinates or pairwise distances). In order to reduce these heterogeneous multi-sets to a fixed-size *feature vector* (input to the discriminator model), we perform a *feature extraction* process which may involve steps such as creating histograms or performing regressions.

In our experiments, our discriminator model outperforms the known (metric-based) algorithms and achieves an average accuracy of 96.48% when identifying

the “better” graph drawing out of a pair. The project source code including the data generation procedure is available online [20].

The remainder of this paper is structured as follows. In Sect. 2 we briefly overview the state-of-the-art in quantifying graph layout aesthetics. Section 4 discusses the used syndromes of aesthetic quality, Sect. 5 feature extraction, and Sect. 6 the discriminator model. The dataset used in our experiments is described in Sect. 7. The results and the comparisons with the known metrics are presented in Sect. 8. Section 9 finalizes the paper and provides an outlook for future work.

2 Related Work

According to empirical studies, graph drawings that maximize one or several quality metrics are more aesthetically pleasing and easier to read [12, 13, 28, 31, 42]. For instance, in their seminal work, Purchase et al. have established [30] that higher numbers of edge crossings and bends as well as lower levels of symmetry negatively influence user performance in graph reading tasks.

Many graph drawing algorithms attempt to optimize multiple quality metrics. As one way to combine them, Huang et al. [13] have used a weighted sum of “simple” metrics, effects of their interactions (see Purchase [29] or Huang and Huang [16]), and error terms to account for possible measurement errors.

In another work, Huang et al. [15] have empirically demonstrated that their “aggregate” metric is sensitive to quality changes and is correlated with the human performance in graph comprehension tasks. They have also noticed that the dependence of aesthetic quality on input quality metrics can be non-linear (e.g. a quadratic relationship better describes the interplay between crossing angles and drawing quality [14]). Our work extends this idea as we allow for arbitrary non-linear dependencies implemented by an artificial neural network.

In evolutionary graph drawing approaches, several techniques have been suggested to “train” a *fitness function*¹ from the user’s responses as a composition of several known quality metrics. Masui [23] modeled the fitness function as a linear combination in which the weights are obtained via genetic programming from the pairs of “good” and “bad” layouts provided by users. The so-called co-evolution was used by Barbosa and Barreto [1] to evolve the weights of the fitness function in parallel with a drawing population in order to match the ranking made by users. Spönemann and others [37] suggested two alternative techniques. In the first one, the user directly chooses the weights with a slider. In the second, they select good layouts from the current population and the weights are adjusted according to the selection. Rosete-Suarez [32] determined the relative importance of individual quality metrics based on user inputs. Several machine learning-based approaches to graph drawing are described by dos Santos Vieira et al. [33]. Recently, Kwon et al. [22] presented a novel work on topological similarity of graphs. Their goal was to avoid expensive computations of graph layouts and their quality measures. The resulting system was able to sketch a graph in different layouts and estimate corresponding quality measures.

¹ Objective function in genetic algorithms that summarizes optimization goals.

3 Definitions

In this paper we consider general simple graphs $G = (V, E)$ where $V = V(G)$ and $E = E(G)$ are the vertex and edge sets of G with $|V| = n$ and $|E| = m$. A *drawing* or *layout* of a graph is its graphical representation where vertices are drawn as points or small circles, and the edges as straight line segments. Vertex positions in a drawing are denoted by $\mathbf{p}^k = (p_1^k, p_2^k)^T$ for $k = 1, \dots, n$ and their set $P = \{\mathbf{p}^k\}_{k=1}^n$. Furthermore, we use $\text{dist}_G(u, v)$ to denote the *graph-theoretical distance* – the length of the shortest path between vertices u and v in G – and $\text{dist}_\Gamma(u, v)$ for the Euclidean distance between u and v in the drawing $\Gamma(G)$.

4 Quality Syndromes of Graph Layouts

A *quality syndrome* of a layout Γ is a multi-set of numbers sharing an interpretation that are known or suspected to correlate with the aesthetic quality (e.g. all pairwise angles between incident edges in Γ). In the following we describe several syndromes (implemented in our code) inspired by popular quality metrics and common statistical tools. The list is by no means exhaustive, nor do we claim syndromes below as necessary or independent. Our model accepts any combination of syndromes; better choices remain to be systematically investigated.

PRINVEC1 and PRINVEC2. The two principal axes of the set P . If we define a covariance matrix $C = \{c_{ij}\}$, $c_{ij} = \frac{1}{n} \sum_{k=1}^n (p_i^k - \bar{p}_i)(p_j^k - \bar{p}_j)$, $i, j \in \{1, 2\}$, where $\bar{p}_i = \frac{1}{n} \sum_{k=1}^n p_i^k$ are the mean values over each dimension, then PRINVEC1 and PRINVEC2 will be its eigenvectors.

PRINCOMP1 and PRINCOMP2. Projections of vertex positions onto $\mathbf{v}_1 = \text{PRINVEC1}$ and $\mathbf{v}_2 = \text{PRINVEC2}$, that is, $\{\langle (\mathbf{p}^j - \bar{\mathbf{p}}), \mathbf{v}_i \rangle\}_{j=1}^n$ for $i \in \{1, 2\}$ where $\langle \cdot, \cdot \rangle$ denotes the scalar product.

ANGULAR. Let $A(v)$ denote the sequence of edges incident to a vertex v , appearing in a clockwise order around it in Γ . Let $\alpha(e_i, e_j)$ denote the clockwise angle between edges e_i and e_j incident to the same vertex. This syndrome is then defined as $\bigcup_{v \in V(G)} \{\alpha(e_i, e_j) : e_i, e_j \text{ are consecutive in } A(v)\}$.

EDGE_LENGTH. $\bigcup_{(u,v) \in E(G)} \{\text{dist}_\Gamma(u, v)\}$ is the set of edge lengths in Γ .

RDF_GLOBAL. $\bigcup_{u \neq v \in V(G)} \{\text{dist}_\Gamma(u, v)\}$ contains distances between all vertices in the drawing. The concept of a *radial distribution function (RDF)* [7] (the distribution of RDF_GLOBAL) is borrowed from Statistical Physics and Crystallography and characterizes the regularity of molecular structures. In large graph layouts it captures regular, periodic and symmetric patterns in the vertex positions.

RDF_LOCAL(d). $\bigcup_{u \neq v \in V(G)} \{\text{dist}_\Gamma(u, v) : \text{dist}(u, v) \leq d\}$ is the set of distances between vertices such that the graph-theoretical distance between them is bounded by $d \in \mathbb{N}$. In our implementation, we compute $\text{RDF_LOCAL}(2^i)$ for $i \in \{0, \dots, \lceil \log_2(D) \rceil\}$ where D is the diameter of G . $\text{RDF_LOCAL}(d)$ in a sense interpolates between **EDGE_LENGTH** ($d = 1$) and **RDF_GLOBAL** ($d \rightarrow \infty$).

TENSION. $\bigcup_{u \neq v \in V(G)} \{\text{dist}_\Gamma(u, v) / \text{dist}_G(u, v)\}$ are the ratios of Euclidean and graph-theoretical distances computed for all vertex pairs. TENSION is motivated by and is related to the well-known stress function [17].

Note that before computing the quality syndromes, we *normalize* all layouts so that the center of gravity of V is at the origin and the mean edge length is fixed in order to remove the effects of scaling and translation (but not rotation).

5 Feature Vectors

The sizes of quality syndromes are in general graph- and layout-dependent. A neural network, however, requires a fixed-size input. A collection of syndromes is condensed to this *feature vector* via *feature extraction*. Our approach to this step relies on several auxiliary definitions. Let $S = \{x_i\}_{i=1}^p$ be a syndrome with p entries. By S^μ we denote the arithmetic mean and by S^ρ the root mean square of S . We also define a *histogram sequence* $S^\beta = \frac{1}{p}(S_1, \dots, S_\beta)$ – normalized counts in a histogram built over S with β bins. The *entropy* [36] of S^β is defined as

$$\mathcal{E}(S^\beta) = - \sum_{i=1}^p \log_2(S_i) S_i. \quad (1)$$

We expect the entropy, as a measure of disorder, to be related to the aesthetic quality of a layout and convey important information to the discriminator.

The entropy $\mathcal{E}(S^\beta)$ is sensitive to the number of bins β (cf. Fig. 1). In order to avoid influencing the results via arbitrary choices of β , we compute it for $\beta = 8, 16, \dots, 512$. After that, we perform a linear regression of $\mathcal{E}(S^\beta)$ as a function of $\log_2(\beta)$. Specifically, we find S^η and S^σ such that $\sum_\beta (S^\sigma \log_2 \beta + S^\eta - \mathcal{E}(S^\beta))^2$ is minimized. The parameters (intercept S^η and slope S^σ) of this regression no longer depend on the histogram size and are used as feature vector components. Figure 1 illustrates that the dependence of $\mathcal{E}(S^\beta)$ on $\log_2(\beta)$ is indeed often close to linear and the regression provides a decent approximation.

A discrete histogram over S can be generalized to a continuous *sliding average*

$$S^F(x) = \frac{\sum_{i=1}^p F(x, x_i)}{\int_{-\infty}^{+\infty} dy \sum_{i=1}^p F(y, x_i)}. \quad (2)$$

A natural choice for the kernel $F(x, y)$ is the Gaussian $F_\sigma(x, y) = \exp\left(-\frac{(x-y)^2}{2\sigma^2}\right)$. By analogy to Eq. 1, we may now define the *differential entropy* [36] as

$$\mathcal{D}(S^{F_\sigma}) = - \int_{-\infty}^{+\infty} dx \log_2(S^{F_\sigma}(x)) S^{F_\sigma}(x). \quad (3)$$

This entropy via kernel function still depends on parameter σ (the filter width). Computing $\mathcal{D}(S^{F_\sigma})$ for multiple σ values as we do for $\mathcal{E}(S^\beta)$ is too expensive. Instead, we have found that using Scott’s Normal Reference Rule [35] as a heuristic to fix σ yields satisfactory results, and allows us to define $S^\epsilon = \mathcal{D}(S^{F_\sigma})$.

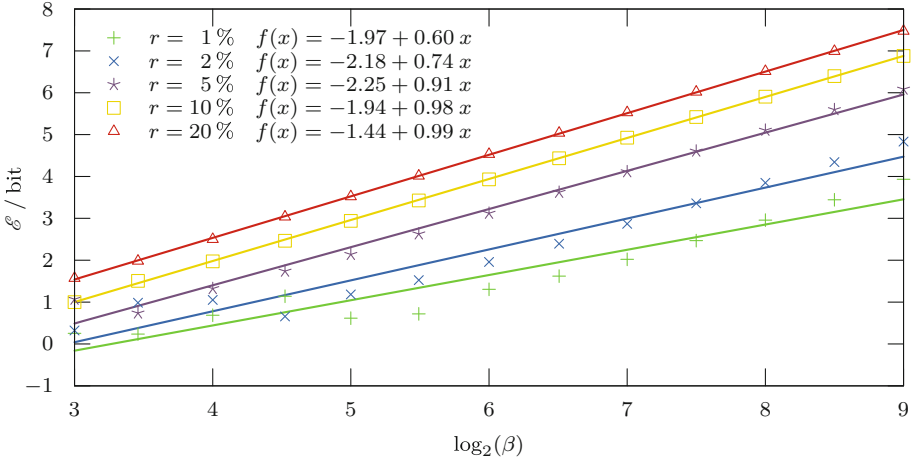


Fig. 1. Entropy $\mathcal{E} = \mathcal{E}(S^\beta)$ computed for histogram sequences S^β defined for different numbers of histogram bins β . Different markers (colors) correspond to several layouts of a regular grid-like graph, progressively distorted according to the parameter r . The dependence of \mathcal{E} on $\log_2(\beta)$ is well approximated by a linear function. Both intercept and slope show a strong correlation with the levels of distortion r . (Color figure online)

Using these definitions, for the most complex syndrome RDF_LOCAL(d) we introduce RDF_LOCAL – a 30-tuple containing the arithmetic mean, root mean square and the differential entropy of RDF_LOCAL(2^i) for $i \in (0, \dots, 9)$. With that², $\text{RDF_LOCAL} = (\text{RDF_LOCAL}(2^i)^\mu, \text{RDF_LOCAL}(2^i)^\rho, \text{RDF_LOCAL}(2^i)^\epsilon)_{i=0}^9$.

Finally, we assemble the 57-dimensional³ feature vector for a layout Γ as

$$F_{\text{layout}}(\Gamma) = \text{PRINVEC1} \cup \text{PRINVEC2} \cup \text{RDF_LOCAL} \cup \bigcup_S (S^\mu, S^\rho, S^\eta, S^\sigma)$$

where S ranges over PRINCOMP1, PRINCOMP2, ANGULAR, EDGE_LENGTH, RDF_GLOBAL and TENSION.

In addition, the discriminator model receives the trivial properties of the underlying graph as the second 2-dimensional vector $F_{\text{graph}}(G) = (\log(n), \log(m))$.

6 Discriminator Model

Feature extractors such as those introduced in the previous section reduce an arbitrary graph G and its arbitrary layout Γ to fixed-size vectors $F_{\text{graph}}(G)$ and

² Values $i < 10$ are sufficient as no graph in our dataset has a diameter exceeding 2^9 .

³ The size is one less than expected from the explanation above because we do not include the arithmetic mean for EDGE_LENGTH as it is constant (due to the layout normalization mentioned earlier) and therefore non-informative.

$F_{\text{layout}}(\Gamma)$. Given a graph G and a pair of its alternative layouts Γ_a and Γ_b , the discriminator function DM receives the feature vectors $\mathbf{v}_a = F_{\text{layout}}(\Gamma_a)$, $\mathbf{v}_b = F_{\text{layout}}(\Gamma_b)$ and $\mathbf{v}_G = F_{\text{graph}}(G)$ and outputs a scalar value

$$t = \text{DM}(\mathbf{v}_G, \mathbf{v}_a, \mathbf{v}_b) \in [-1, 1]. \quad (4)$$

The interpretation is as follows: if $t < 0$, then the model believes that Γ_a is “prettier” than Γ_b ; if $t > 0$, then it prefers Γ_b . Its magnitude $|t|$ encodes the confidence level of the decision (the higher $|t|$, the more solid the answer).

For the implementation of the function DM we have chosen a practically convenient and flexible model structure known as *Siamese neural networks*, originally proposed by Bromley and others [3] that is defined as

$$\text{DM}(\mathbf{v}_G, \mathbf{v}_a, \mathbf{v}_b) = \text{GM}(\boldsymbol{\sigma}_a - \boldsymbol{\sigma}_b, \mathbf{v}_G) \quad (5)$$

where $\boldsymbol{\sigma}_a = \text{SM}(\mathbf{v}_a)$ and $\boldsymbol{\sigma}_b = \text{SM}(\mathbf{v}_b)$. The *shared model* SM and the *global model* GM are implemented as multi-layer neural networks with a simple structure shown in Fig. 2. The network was implemented using the *Keras* [18] framework with the *TensorFlow* [40] library as back-end.

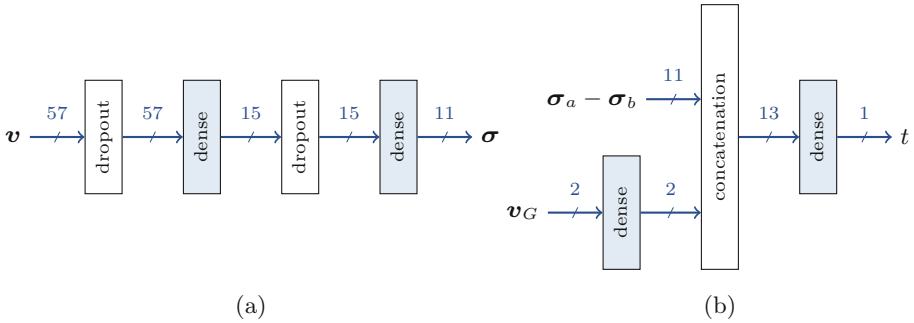


Fig. 2. Structure of the neural networks $\text{SM}(\mathbf{v})$ (a) and $\text{GM}(\boldsymbol{\sigma}_a - \boldsymbol{\sigma}_b, \mathbf{v}_G)$ (b). Shaded blocks denote standard network layers, and the numbers on the arrows denote the dimensionality of the respective representations.

The SM network (Fig. 2(a)) consists of two “dense” (fully-connected) layers, each preceded by a “dropout” layer (discarding 50% and 25% of the signals, respectively). Dropout is a stochastic regularization technique intended to avoid overfitting that was first proposed by Srivastava and others [38].

In the GM network (Fig. 2(b)), the graph-related feature vector \mathbf{v}_G is passed through an auxiliary dense layer, and concatenated with the difference signal $(\boldsymbol{\sigma}_a - \boldsymbol{\sigma}_b)$ obtained from the output vectors of SM for the two layouts. The final dense layer produces the scalar output value. The first and the auxiliary layers use linear activation functions, the hidden layer uses ReLU [11] and the final layer hyperbolic tangent activation. Following the standard practice, the

inputs to the network are normalized by subtracting the mean and dividing by the standard deviation of the feature computed over the complete dataset.

In total, the DM model has 1 066 free parameters, trained via stochastic gradient descent-based optimization of the mean squared error (MSE) loss function.

7 Training and Testing Data

For training, all machine learning methods require datasets representing the variability of possible inputs. Our DM model needs a dataset containing graphs, their layouts, and known aesthetic orderings of layout pairs. We have assembled such a dataset using two types of sources. First, we used the collections of the well-known graph archives **ROME**, **NORTH** and **RANDDAG** which are published on graphdrawing.org as well as the NIST’s “Matrix Market” [2].

Second, we have generated random graphs using the algorithms listed below. As a by-product, some of them produce layouts that stem naturally from the generation logic. We refer to these as *native* layouts (see [19] for details).

GRID. Regular $n \times m$ grids. Native layouts: regular rectangular grids.

TORUS1. Same as **GRID**, but the first and the last “rows” are connected to form a 1-torus (a cylinder). No native layouts.

TORUS2. Same as **TORUS1**, but also the first and the last “columns” are connected to form a 2-torus (a doughnut). No native layouts.

LINDENMAYER. Uses a stochastic L-system [27] to derive increasingly complex graphs by performing random replacements of individual vertices with more complicated substructures such as an n -ring or an n -clique. Produces a planar native layout.

QUASI $\langle n \rangle$ D for $n \in \{3, \dots, 6\}$. Projection of a primitive cubic lattice in an n -dimensional space onto a 2-dimensional plane intersecting that space at a random angle. The native layout follows from the construction.

MOSAIC1. Starts with a regular polygon and randomly divides faces according to a set of simple rules until the desired graph size is reached. The rules include adding a vertex connected to all vertices of the face; subdividing each edge and adding a vertex that connects to each subdivision vertex; subdividing each edge and connecting them to a cycle. The native layout follows from the construction.

MOSAIC2. Applies a randomly chosen rule of **MOSAIC1** to every face, with the goal of obtaining more symmetric graphs.

BOTTLE. Constructs a graph as a three-dimensional mesh over a random solid of revolution. The native layout is an axonometric projection.

For each graph, we have computed force-directed layouts using the FM^3 [10] and stress-minimization [17] algorithms. We assume these and native layouts to be generally aesthetically pleasing and call them all *proper* layouts of a graph.

Furthermore, we have generated *a priori* un-pleasing (*garbage*) layouts as follows. Given a graph $G = (V, E)$, we generate a random graph $G' = (V', E')$ with $|V'| = |V|$ and $|E'| = |E|$ and compute a force-directed layout for G' .

The coordinates found for the vertices V' are then assigned to V . We call these “phantom” layouts due to the use of a “phantom” graph G' . We find that phantom layouts look less artificial than purely random layouts when vertex positions are sampled from a uniform or a normal distribution. This might be due to the fact that G and G' have the same density and share some beneficial aspects of the force-directed method (such as mutual repelling of nodes).

For training and testing of the discriminator model we need a corpus of labeled pairs – triplets (Γ_a, Γ_b, t) where Γ_a and Γ_b are two different layouts for the same graph and $t \in [-1, 1]$ is a value indicating the relative aesthetic quality of Γ_a and Γ_b . A negative (positive) value for t expresses that the quality of Γ_a is superior (inferior) compared to Γ_b and the magnitude of t expresses the confidence of this prediction. We only use pairs with sufficiently large $|t|$.

As manually-labelled data were unavailable, we have fixed the values of t as follows. First, we paired a proper and a garbage layout of a graph. The assumption is that the former is always more pleasing (i.e. $t = \pm 1$). Second, in order to obtain more nuanced layout pairs and to increase the amount of data, we have employed the well-known technique of *data augmentation* as follows.

Layout Worsening: Given a proper layout Γ , we apply a transformation designed to gradually reduce its aesthetic quality that is modulated by some parameter $r \in [0, 1]$, resulting in a transformed layout Γ'_r . By varying the degree r of the distortion, we may generate a sequence of layouts ordered by their anticipated aesthetic value: a layout with less distortion is expected to be more pleasing than a layout with more distortion when starting from a presumably decent layout. We have implemented the following worsening techniques. **PERTURB:** add Gaussian noise to each node’s coordinates. **FLIP_NODES:** swap coordinates of randomly selected node pairs. **FLIP_EDGES:** same as **FLIP_NODES** but restricted to connected node pairs. **MOVLSQ:** apply an affine deformation based on moving least squares suggested (although for a different purpose) by Schaefer et al. [34]. In essence, all vertices are shifted according to some smoothly varying coordinate mapping.

Layout Interpolation: As the second data augmentation technique, we linearly interpolated the positions of corresponding vertices between the proper and garbage layouts of the same graph. The resulting label t is then proportional to the difference in the interpolation parameter.

In total, using all the methods described above, we have been able to collect a database of about 36 000 labeled layout pairs.

8 Evaluation

The performance of the discriminator model was evaluated using *cross-validation* with 10-fold random subsampling [21]. In each round, 20% of graphs (with all their layouts) were chosen randomly and were set aside for testing, and the model was trained using the remaining layout pairs. Of N labeled pairs used for testing,

in each round we computed the number N_{correct} of pairs for which the model properly predicted the aesthetic preference, and derived the accuracy (success rate) $A = N_{\text{correct}}/N$. The standard deviation of A over the 10 runs was taken as the uncertainty of the results. With the average number of test samples of $N = 7415$, the eventual success rate was $\mathbf{A} = (96.48 \pm 0.85)\%$.

8.1 Comparison with Other Metrics

In order to assess the relative standing of the suggested method, we have implemented two known aesthetic metrics (*stress* and the *combined metric* by Huang et al. [15]) and evaluated them over the same dataset. The metric values were trivially converted to the respective discriminator function outputs.

Stress \mathcal{T} of a layout Γ of a simple connected graph $G = (V, E)$ was defined by Kamada and Kawai [17] as

$$\mathcal{T}(\Gamma) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n k_{ij} (\text{dist}_{\Gamma}(v_i, v_j) - L \text{dist}_G(v_i, v_j))^2, \quad (6)$$

where L denotes the desirable edge length and $k_{ij} = K/\text{dist}_G(v_i, v_j)^2$ is the strength of a “spring” attached to v_i and v_j . The constant K is irrelevant in the context of discriminator functions and can be set to any value.

As observed by Welch and Kobourov [43], the numeric value of stress depends on the layout scale via the constant L in the Eq. 6 which complicates comparisons. Their suggested solution was for each layout to find L that minimizes \mathcal{T} (e.g. using binary search). In our implementation, we applied a similar technique based on fitting and minimizing a quadratic function to the stress computed at three scales. We refer to this quantity as **STRESS**.

The combined metric proposed by Huang et al. [15] (referred to as **COMB**) is a weighted average of four simpler quality metrics: the number of edge crossings (**CC**), the minimum crossing angle between any two edges in the drawing (**CR**), the minimum angle between two adjacent edges (**AR**), and the standard deviation computed over all edge lengths (**EL**).

The average is computed over the so-called z -scores of the above metrics. Each z -score is found by subtracting the mean and dividing by the standard deviation of the metric for all layouts of a given graph to be compared with each other. More formally, let G be a graph and $\Gamma_1, \dots, \Gamma_k$ be its k layouts to be compared pairwise. Let $M(\Gamma_i)$ be the value of metric M for Γ_i and μ_M and σ_M be the mean and the standard deviation of $M(\Gamma_i)$ for $i \in \{1, \dots, k\}$. Then

$$z_M^{(i)} = \frac{M(\Gamma_i) - \mu_M}{\sigma_M} \quad (7)$$

is the z -score for metric M and layout Γ_i . The combined metric then is

$$\text{COMB}(\Gamma_j) = \sum_M w_M z_M^{(j)}. \quad (8)$$

The weights w_M were found via Nelder-Mead maximization [26] of the prediction accuracy over the training dataset⁴.

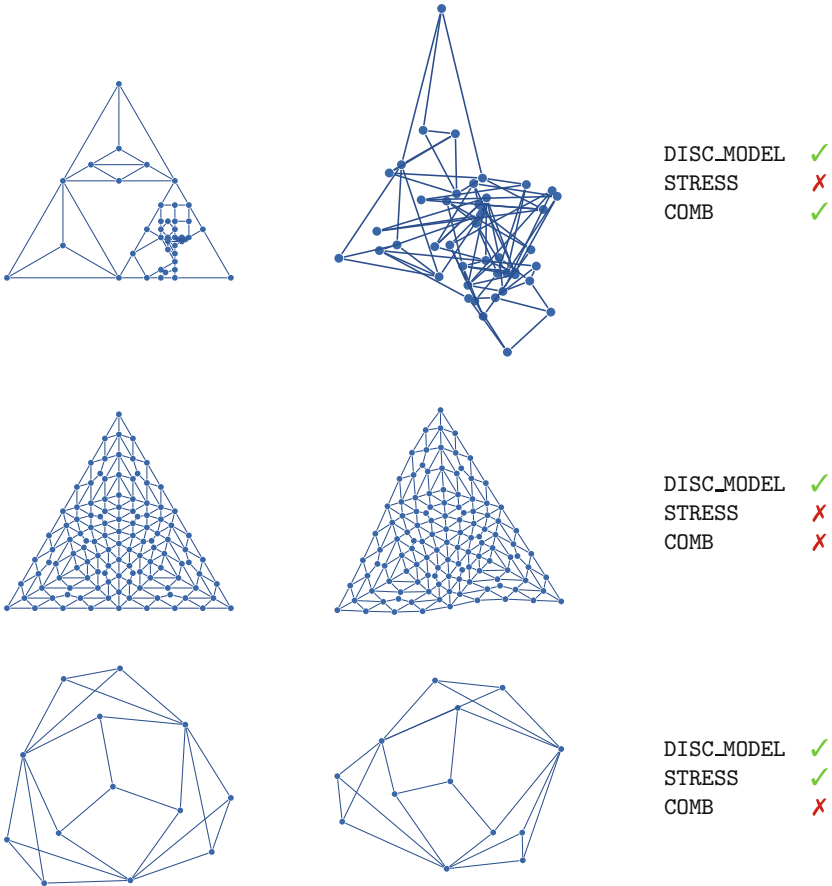


Fig. 3. Examples where our discriminator model (DISC_MODEL) succeeds (✓) and the competing metrics fail (✗) to predict the answer correctly. In each row, the layout on the left is expected to be superior compared to the one on the right.

The accuracy of the stress-based and the combined model-based discriminators is shown in Table 1. In most cases, our model outperforms these algorithms by a comfortable margin. Figure 3 provides examples of mis-predictions. By inspecting such cases, we notice that STRESS often fails to guess the aesthetics of (almost) planar layouts that contain both very short and very long edges (such behavior may also be inferred from the definition of STRESS). We observe

⁴ The obtained weights are: $w_{EL} = +0.4803 \pm 0.0855$, $w_{CC} = +0.4679 \pm 0.1069$, $w_{CR} = -0.0431 \pm 0.0315$, $w_{AR} = -0.0087 \pm 0.0072$.

that there are planar graphs, such as nested triangulations, for which this property is unavoidable in planar drawings. The mis-predictions of COMB seem to be due to the high weight of the edge length metric EL. Both STRESS and COMB are weaker than our model in capturing the absolute symmetry and regularity of layouts.

Table 1. Accuracy scores for the COMB and STRESS model. The standard deviation in each column is estimated based on the 5-fold cross-validation (using 20% of data for testing each time). The “Advantage” column shows the improvement in the accuracy of our model with respect to the alternative metric.

Metric	Success rate	Advantage
STRESS	$(93.49 \pm 0.86)\%$	$(2.99 \pm 1.01)\%$
COMB	$(92.76 \pm 1.03)\%$	$(3.71 \pm 1.22)\%$

8.2 Significance of Individual Syndromes

In order to estimate the influence of individual syndromes on the final result, we have tested several modifications of our model. For each syndrome, we considered the case when the feature vector contained only that syndrome. In the second case, that syndrome was removed from the original feature vector. The entries for the omitted features were set to zero. The results are shown in Table 2.

Table 2. Success rates of our discriminator when a syndrome is excluded from the feature vector, and when the feature vector contains only that a syndrome. Note that RDF_LOCAL is a family of syndromes that are all included or excluded together. The apparent paradox of higher success rates when some syndromes are excluded can be explained by a statistical fluctuation and is well within the listed range of uncertainty.

Property	Sole exclusion	Sole inclusion
PRINCOMP1	$(96.37 \pm 0.84)\%$	$(55.51 \pm 6.50)\%$
PRINCOMP2	$(96.20 \pm 0.76)\%$	$(61.08 \pm 5.24)\%$
EDGE_LENGTH	$(96.33 \pm 0.59)\%$	$(71.65 \pm 3.38)\%$
ANGULAR	$(96.40 \pm 0.34)\%$	$(77.79 \pm 6.06)\%$
RDF_GLOBAL	$(95.92 \pm 0.94)\%$	$(86.37 \pm 3.43)\%$
TENSION	$(96.83 \pm 0.31)\%$	$(89.78 \pm 0.95)\%$
RDF_LOCAL	$(90.04 \pm 2.04)\%$	$(94.78 \pm 1.60)\%$
Baseline using all properties	$(96.48 \pm 0.85)\%$	

As can be observed, the dominant contribution to the accuracy of the model is due to the RDF-based properties RDF_LOCAL and RDF_GLOBAL. The exclusion

of other syndromes does not significantly change the results (they agree within the estimated uncertainty). However, the sole inclusion of these syndromes still performs better than random choice. This suggests that there is a considerable overlap between the aesthetic aspects captured by various syndromes. Further analysis is needed to identify the nature and the magnitude of these correlations.

9 Conclusion

In this paper we propose a machine learning-based discriminator model that selects the more aesthetically pleasing drawing from a pair of graph layouts. Our model picks the “better” layout in more than 96% cases and outperforms known stress-based and linear combination-based models. To the best of our knowledge, this is the first application of machine learning methods to this question. Previously, such techniques have proven successful in a range of complex issues involving aesthetics, prior knowledge, and unstated rules in object recognition, industrial design, and digital arts. As our model uses a simple network architecture, investigating the performance of more complex networks is warranted.

Previous efforts were focused on determining the aesthetic quality of a layout as a weighted average of individual quality metrics. We extend these ideas and findings in the sense that we do not assume any particular form of dependency between the overall aesthetic quality and the individual quality metrics.

Going beyond simple quality metrics, we define quality syndromes that capture arrays of information about graphs and layouts. In particular, we borrow the notion of RDF from Statistical Physics and Crystallography; RDF-based features demonstrate the strongest potential in extracting the aesthetic quality of a layout. We expect RDFs (describing the microscopic structure of materials) to be the most relevant for large graphs. It is tempting to investigate whether further tools from physics can be useful in capturing drawing aesthetics.

From multiple syndromes, we construct fixed-size feature vectors using common statistical tools. Our feature vector does not contain any information on crossings or crossing angles, nevertheless its performance is superior with respect to the weighted averages-based model which accounts for both. It would be interesting to investigate whether including these and other features further improves the performance of the neural network-based model.

In order to train and evaluate the model, we have assembled a relatively large corpus of labeled pairs of layouts, using available and generated graphs and exploiting the assumption that layouts produced by force-directed algorithms and native graph layouts are aesthetically pleasing and that disturbing them reduces the aesthetic quality. We admit that this study should ideally be repeated with human-labeled data. However, this requires that a dataset be collected with a size similar to ours, which is a challenging task. Creating such a dataset may become a critically important accomplishment in the graph drawing field.

References

1. Barbosa, H.J.C., Barreto, A.M.S.: An interactive genetic algorithm with coevolution of weights for multiobjective problems. In: Spector, L., Goodman, E.D., Wu, A., Langdon, W.B., Voigt, H.M. (eds.) *An Interactive Genetic Algorithm With Co-evolution of Weights for Multiobjective Problems*, GECCO 2001, pp. 203–210. Morgan Kaufmann Publishers Inc. (2001)
2. Boisvert, R.F., Pozo, R., Remington, K., Barrett, R.F., Dongarra, J.J.: Matrix market: a web resource for test matrix collections. In: Boisvert, R.F. (ed.) *Quality of Numerical Software: Assessment and enhancement*, pp. 125–137. Springer, Heidelberg (1997). https://doi.org/10.1007/978-1-5041-2940-4_9
3. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a “Siamese” time delay neural network. In: *Advances in Neural Information Processing Systems*, pp. 737–744 (1994). <https://doi.org/10.1142/S0218001493000339>
4. Dev, K., Villar, N., Lau, M.: Polygons, points, or voxels?: Stimuli selection for crowdsourcing aesthetics preferences of 3d shape pairs. In: Gooch, B., Gingold, Y.I., Winnemoeller, H., Bartram, L., Spencer, S.N. (eds.) *Proceedings of the symposium on Computational Aesthetics, CAE 2017, Los Angeles, California, USA*, pp. 2:1–2:7. ACM (2017). <https://doi.org/10.1145/3092912.3092918>
5. Eades, P.: A heuristic for graph drawing. *Congr. Numer.* **24**, 149–160 (1984)
6. Eades, P., Hong, S., Nguyen, A., Klein, K.: Shape-based quality metrics for large graph visualization. *J. Graph Algorithms Appl.* **21**(1), 29–53 (2017). <https://doi.org/10.7155/jgaa.00405>
7. Findenegg, G.H., Hellweg, T.: *Statistische Thermodynamik*, 2nd edn. Springer, Darmstadt (2015). <https://doi.org/10.1007/978-3-642-37872-0>
8. Fredrickson, B.L.: What good are positive emotions. *Rev. Gen. Psychol.* **2**, 300–319 (1998)
9. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. *Softw. Pract. Exper.* **21**(11), 1129–1164 (1991). <https://doi.org/10.1002/spe.4380211102>
10. Hachul, S., Jünger, M.: Drawing large graphs with a potential-field-based multilevel algorithm. In: Pach, J. (ed.) *GD 2004. LNCS*, vol. 3383, pp. 285–295. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31843-9_29
11. Hahnloser, R.H.R., Sarpeshkar, R., Mahowald, M.A., Douglas, R.J., Seung, H.S.: Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature* **405**, 947–951 (2000). <https://doi.org/10.1038/35016072>
12. Huang, W., Eades, P.: How people read graphs. In: Hong, S. (ed.) *Asia-Pacific Symposium on Information Visualisation, APVIS 2005*, pp. 51–58, Australia, Sydney (2005)
13. Huang, W., Eades, P., Hong, S.H., Lin, C.C.: Improving multiple aesthetics produces better graph drawings. *J. Vis. Lang. Comput.* **24**(4), 262–272 (2013). <https://doi.org/10.1016/j.jvlc.2011.12.002>
14. Huang, W., Hong, S., Eades, P.: Effects of crossing angles. In: *IEEE VGTC Pacific Visualization Symposium 2008, PacificVis 2008*, Kyoto, Japan, pp. 41–46 (2008). <https://doi.org/10.1109/PACIFICVIS.2008.4475457>
15. Huang, W., Huang, M.L., Lin, C.C.: Evaluating overall quality of graph visualizations based on aesthetics aggregation. *Inf. Sci.* **330**, 444–454 (2016). <https://doi.org/10.1016/j.ins.2015.05.028>, SI: Visual Info Communication

16. Huang, W., Huang, M.: Exploring the relative importance of crossing number and crossing angle. In: Dai, G., et al. (eds.) Proceedings of the 3rd International Symposium on Visual Information Communication, VINCI 2010, pp. 1–8, ACM (2010). <https://doi.org/10.1145/1865841.1865854>
17. Kamada, T., Kawai, S.: An algorithm for drawing general undirected graphs. *Inf. Process. Lett.* **31**(1), 7–15 (1989). [https://doi.org/10.1016/0020-0190\(89\)90102-6](https://doi.org/10.1016/0020-0190(89)90102-6)
18. Keras. <https://keras.io/>
19. Klammler, M.: Aesthetic value of graph layouts: investigation of statistical syndromes for automatic quantification. Master's thesis, Karlsruhe Institute of Technology (2018). <http://klammler.eu/msc/>
20. Klammler, M., et al.: Source code for aesthetic discrimination of graph layouts. <https://github.com/5gon12eder/msc-graphstudy>
21. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI 1995, Montréal Québec, Canada, vol. 2, pp. 1137–1145. Morgan Kaufmann (1995)
22. Kwon, O.H., Crnovršanin, T., Ma, K.L.: What would a graph look like in this layout? A machine learning approach to large graph visualization. *IEEE Trans. Vis. Comput. Graph.* **24**(1), 478–488 (2018). <https://doi.org/10.1109/TVCG.2017.2743858>
23. Masui, T.: Evolutionary learning of graph layout constraints from examples. In: Szekely, P.A. (ed.) Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology, pp. 103–108. UIST 1994. ACM (1994). <https://doi.org/10.1145/192426.192468>
24. Nishiyama, M., Okabe, T., Sato, Y., Sato, I.: Sensation-based photo cropping. In: Gao, W., et al. (eds.) Proceedings of the 17th ACM International Conference on Multimedia. pp. 669–672. MM 2009, ACM (2009). <https://doi.org/10.1145/1631272.1631384>
25. Norman, D.A.: Emotion & design: attractive things work better. *Interactions* **9**(4), 36–42 (2002). <https://doi.org/10.1145/543434.543435>
26. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical Recipes: The Art of Scientific Computing, 3 edn. Cambridge University Press (2007)
27. Prusinkiewicz, P., Lindenmayer, A.: The Algorithmic Beauty of Plants. Springer, New York (1990). <https://doi.org/10.1007/978-1-4613-8476-2>
28. Purchase, H.: Which aesthetic has the greatest effect on human understanding? In: DiBattista, G. (ed.) GD 1997. LNCS, vol. 1353, pp. 248–261. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63938-1_67
29. Purchase, H.C.: Performance of layout algorithms: comprehension, not computation. *J. Vis. Lang. Comput.* **9**(6), 647–657 (1998). <https://doi.org/10.1006/jvlc.1998.0093>
30. Purchase, H.C., Cohen, R.F., James, M.: Validating graph drawing aesthetics. In: Brandenburg, F.J. (ed.) GD 1995. LNCS, vol. 1027, pp. 435–446. Springer, Heidelberg (1996). <https://doi.org/10.1007/BFb0021827>
31. Purchase, H.C., Hamer, J., Nöllenburg, M., Kobourov, S.G.: On the usability of Lombardi graph drawings. In: Didimo, W., Patrignani, M. (eds.) GD 2012. LNCS, vol. 7704, pp. 451–462. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36763-2_40
32. Rosete-Suarez, A., Sebag, M., Ochoa-Rodriguez, A.: A study of evolutionary graph drawing: laboratoire de Recherche en Informatique (LRI), Université Paris-Sud XI, p. 1228. Technical report (1999)

33. dos Santos Vieira, R., do Nascimento, H.A.D., da Silva, W.B.: The application of machine learning to problems in graph drawing – a literature review. In: Proceedings of the 7th International Conference on Information, Process, and Knowledge Management, eKNOW 2015, Lisbon, Portugal, pp. 112–118 (2015)
34. Schaefer, S., McPhail, T., Warren, J.: Image deformation using moving least squares. *ACM Trans. Graph.* **25**(3), 533–540 (2006). <https://doi.org/10.1145/1141911.1141920>
35. Scott, D.W.: On optimal and data-based histograms. *Biometrika* **66**(3), 605–610 (1979). <https://doi.org/10.1093/biomet/66.3.605>
36. Shannon, C.E.: A mathematical theory of communication. *Bell Syst. Tech. J.* **27**(4), 623–656 (1948). <https://doi.org/10.1002/j.1538-7305.1948.tb00917.x>
37. Spönemann, M., Duderstadt, B., von Hanxleden, R.: Evolutionary meta layout of graphs. In: Dwyer, T., Purchase, H., Delaney, A. (eds.) *Diagrams 2014*. LNCS (LNAI), vol. 8578, pp. 16–30. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44043-8_3
38. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
39. Tamassia, R.: *Handbook of Graph Drawing and Visualization*. Discrete Mathematics and Its Applications. CRC Press (2013)
40. Tensor flow. <https://tensorflow.org/>
41. Tractinsky, N., Katz, A.S., Ikar, D.: What is beautiful is usable. *Interact. Comput.* **13**(2), 127–145 (2000). [https://doi.org/10.1016/S0953-5438\(00\)00031-X](https://doi.org/10.1016/S0953-5438(00)00031-X)
42. Ware, C., Purchase, H.C., Colpoys, L., McGill, M.: Cognitive measurements of graph aesthetics. *Inf. Vis.* **1**(2), 103–110 (2002). <https://doi.org/10.1057/palgrave.ivs.9500013>
43. Welch, E., Kobourov, S.: Measuring symmetry in drawings of graphs. *Comput. Graph. Forum* **36**(3), 341–351 (2017). <https://doi.org/10.1111/cgf.13192>