A Quality Metric for Symmetric Graph Drawings

Amyra Meidiana, Seok-Hee Hong, Peter Eades, Daniel Keim

Abstract
Symmetry is an important aesthetic criteria in graph drawing and network visualisation. Symmetric graph drawings aim to faithfully represent automorphisms of graphs as geometric symmetries in a drawing.

In this paper, we design and implement a framework for quality metrics that measure symmetry, that is, how faithfully a drawing of a graph displays automorphisms as geometric symmetries. The quality metrics are based on geometry (i.e., Euclidean distance) as well as mathematical group theory (i.e., orbits of automorphisms).

More specifically, we define two varieties of symmetry quality metrics: (1) for displaying a single automorphism as a symmetry (axial or rotational) and (2) for displaying a group of automorphisms (cyclic or dihedral). We also present algorithms to compute the symmetric quality metrics.

We validate our symmetric quality metrics using deformation experiments. We then use the metrics to evaluate a number of established graph drawing layouts to compare how faithfully they display automorphisms of a graph as geometric symmetries.

Index Terms: Human-centered computing [Visualization]: Visualization design and evaluation methods

1 Introduction

Graph drawing aims to construct a visually-informative drawing of an abstract graph in the plane. Symmetry is one of the most important aesthetic criteria that represent the structure and properties of a graph visually. Symmetric graph drawings aim to faithfully represent automorphisms of graphs as geometric symmetries in the drawing. Also, a symmetric drawing of a graph enables an understanding of the entire graph to be built up from that of a smaller subgraph.

Symmetric drawings of a graph G are clearly related to the automorphisms of G, and algorithms for constructing symmetric drawings have two steps:

1. Find the “appropriate” automorphisms, and
2. Draw the graph displaying these automorphisms as symmetries.

The problem of determining whether a graph has a nontrivial automorphism is automorphism complete. However, the problem of determining whether a graph has a nontrivial geometric automorphism is NP-complete. Linear time algorithms to construct symmetric drawings for restricted classes of graphs exist (e.g., trees, outerplanar graphs, series-parallel digraphs, and planar graphs). For general graphs, heuristics and exact algorithms are available.

It is important to ensure that a symmetric graph drawing accurately represent the automorphisms of the underlying graph to the greatest extent possible. However, existing symmetry detection and quality metrics for graph drawings do not focus on this comparison between detected geometric symmetry in a graph drawing and the automorphisms of the graph.

In this paper, we design and implement a framework for quality metrics for graph drawings that measure symmetry, that is, how faithfully a drawing of a graph displays automorphisms as geometric symmetries. The quality metrics are based on geometry (i.e., Euclidean distance) as well as mathematical group theory (i.e., orbits of automorphisms).

More specifically, we present the following contributions:

1. We design and implement a framework for a quality metric for symmetric graph drawing, which measures the symmetry quality of a drawing of a graph based on the comparison between the symmetry of the drawing and the automorphism of the graph. We define two varieties of the symmetry quality metrics, one to measure how well a graph drawing displays a single automorphism as a symmetry (rotational or axial) and one to measure how well a graph drawing displays a group of automorphisms simultaneously (cyclic or dihedral groups).

2. We present algorithms to compute the symmetry quality metrics in O(n log n) time for rotational and axial symmetry and O(kn log n) for automorphism groups, where n is the number of vertices in the graph and k is the size of the automorphism group.

3. We validate the single automorphism detection version of the symmetry quality metrics through deformation experiments of graph drawings, showing that the scores computed by our metric decrease when the drawings are distorted further from exact symmetry.

4. We validate the automorphism group detection version of the symmetry quality metrics through comparing drawings displaying different groups of automorphisms, showing that our metric effectively captures the difference in symmetry quality between drawings that display different numbers of automorphisms as symmetries.

5. We use our metric to compare a number of established graph drawing layouts to compare how faithfully they display automorphisms of a graph as geometric symmetries. We confirm the effectiveness of the concentric circles layout in displaying a graph’s automorphisms as symmetries.

2 Related Works

2.1 Symmetries and geometric automorphisms

An automorphism of a graph is a permutation of the vertices that preserves the adjacency of each vertex. Suppose that a permutation group A acts on a set V. We say that \( \phi \in A \) fixes \( v \in V \) if \( \phi(v) = v \); if \( \phi \) fixes \( v \) for every \( \phi \in A \) then \( A \) fixes \( v \). If \( V' \subseteq V \) and \( \phi(v') \in V' \) for all \( v' \in V' \) then \( \phi \) restricts \( V' \) setwise. A subset \( V' \) of \( V \) is an orbit of \( A \) if, for each \( u, v \in V' \), there is \( \phi \in A \) such that \( \phi(u) = v \) and \( V' \) does not contain a nonempty subset with this property.

< ![Konstanzer Online-Publikations-System (KOPS)](http://nbn-resolving.de/urn:nbn:de:bz:352-2-1or2cdmtya1no3)>
It is important to use a rigorous model, introduced by Eades and Lin [9], for the intuitive concept of symmetry display.

The symmetries of a set of points in the plane (such as a two-dimensional graph drawing) form a group called the symmetry group of the set. A symmetry \( \sigma \) of a drawing \( D \) of a graph \( G \) induces an automorphism of \( G \), in that the restriction of \( \sigma \) to the points representing vertices of \( G \) is an automorphism of \( G \). A drawing \( D \) of a graph \( G \) displays an automorphism \( \phi \) of \( G \) if there is symmetry \( \sigma \) of \( D \) which induces \( \phi \). The symmetry group of a graph drawing induces an automorphism group of the graph. An automorphism group \( A \) of a graph \( G \) is a geometric automorphism group if there is a drawing of \( G \) which displays every element of \( A \).

A non-trivial symmetry of a finite set of points in the plane is either a rotation about a point or a reflection about a line. A geometric automorphism is a rotational automorphism (respectively reflectional/axial automorphism) if it is induced by a rotation (respectively reflection). Eades and Lin [9] showed that a nontrivial geometric automorphism group is one of three kinds:

1. a group of size 2 generated by an axial automorphism;
2. a cyclic group of size \( k \) generated by a rotational automorphism;
3. a dihedral group of size \( 2k \) generated by a rotational automorphism of order \( k \) and an axial automorphism.

### 2.2 Symmetric Graph Drawing

In general, determining whether a graph can be drawn symmetrically in two dimensions is NP-complete [19]. Exact algorithms based on Branch and Cut [5] and group theory [1] are available.

Linear-time algorithms are available for symmetric drawings of limited classes of graphs, such as trees [21], outerplanar graphs [22], and series-parallel digraphs [15]. Linear-time algorithms have also been presented for maximally symmetric drawings of triconnected [15], biconnected [12], oneconnected [13], and disconnected planar graphs [14]. For a survey on symmetric drawings of graphs in two dimensions, see [8].

### 2.3 Graph Drawing Symmetry Quality Metrics

Purchase [24] defined a metric measuring the symmetry of a graph drawing by checking, for each pair of vertices, whether there is a symmetric subgraph around the pair, calculating a weighted symmetry value of the symmetric subgraph if it exists, and adding the weighted symmetry value of all symmetric subgraphs. However, this metric has a best runtime of \( O(n^2) \) and only considers axial symmetry.

Klapaukh et al. [17] defined a metric which detects rotational, axial, and translational symmetry in a node-link graph drawing, using methods adapted from computer vision. Taking as input an image of a drawing of a graph, this method detects symmetries by converting edges detected in the image into line vectors and then computing the symmetry quality for each line and each pair of lines, giving the runtime as \( O(m^2) \) time in the number of edges, with the worst case of \( O(n^2) \) time in the number of vertices for dense graphs.

### 2.4 Geometric Symmetry Detection

Optimal algorithms for exact symmetry detection for two-dimensional point sets and polygons run in \( O(n \log n) \) time for detecting rotational or axial symmetry in two-dimensional point sets [21][25]. These algorithms work by sorting the points by their angle around and distance from a centroid, then finding a palindromic sequence of angles and distances (for axial symmetry) or a subsequence that is repeated within the full sequence (for rotational symmetry). However, these algorithms only give a binary answer to whether a point set displays a symmetry and are not suitable to quantify the approximate symmetry of a point set.

Zabrodsky et al. [27] introduce the symmetry distance method, providing another method to quantify the approximate symmetry of a geometric object. This method takes as input a point set \( P \) and a symmetry \( \sigma \) to be checked and computes a symmetric point set \( P_\sigma \) realizing the input symmetry. This is done through a “folding” transform method that “folds” points in the same orbit, averages their positions, and “unfolds” them into a symmetric configuration. This method minimizes the Euclidean distance between each point in \( P \) and its image in \( P_\sigma \), and the symmetry distance is given as the average of these Euclidean distances for every point in \( P \).

### 3 Symmetry Quality Metric Framework

We propose a new quality metric for graph visualization, the symmetry quality metric, for measuring how well the drawing of a graph displays selected geometric automorphisms of the underlying graph. Our metric is a faithfulness metric comparing the geometric symmetry detected in a drawing of a graph with the graph’s automorphisms, the ground truth information, unlike existing metrics which only attempts to detect symmetry from the drawing. Figure 1 summarizes the framework used for our proposed metric.

Let \( G \) be a graph and \( \phi \) be a automorphism of \( G \). It is important that \( \phi \) be a geometric automorphism of \( G \), as otherwise it is impossible to display it as a symmetry of a drawing of \( G \) [9]. The framework computes the symmetry quality score using the following steps:

**Framework: Symmetry Quality Metric**

**Step 1:** Apply a layout algorithm to \( G \) to obtain a graph drawing \( D \), which provides geometric positions for each vertex in \( G \).

**Step 2:** Detect a geometric symmetry of \( D \), obtaining symmetry \( \sigma \).

**Step 3:** Compare \( \sigma \) to \( \phi \) to compute the symmetry quality metric.

### 3.1 Symmetry Quality Metric for a Single Automorphism

Given a drawing of a graph with an exact symmetry, a geometric symmetry detection algorithm can be used to detect a symmetry of the drawing, which induces an automorphism of the graph, and this result can be compared to the input automorphism. However, in practice, automatic graph layout algorithms may produce drawings that are not perfectly symmetric. We therefore define a refinement that uses approximate symmetry detection to quantify how far the drawing is from an exact symmetry that displays an input automorphism.

The approximate symmetry of a point set \( P \) can be defined as a Euclidean distance between it and a symmetric point set \( P_\sigma \). In
this refinement of our metric, for each orbit of \( \phi \), we compute the Euclidean distance needed to transform the position of its points in \( D \) to perfect symmetry, and use this distance to compute the symmetry quality.

We define two different formulas to compute the symmetry quality metric, \( SQ \), given the number of symmetric orbits and the Euclidean distance to exact symmetry for the asymmetric orbits. The first formula gives equal weight to the proportion of symmetric orbits and the average Euclidean distance to perfect symmetry of asymmetric orbits:

\[
SQ_1 = \frac{1}{2} \left( \frac{|O_{\text{sym}}|}{|O|} + \frac{1}{|O_{\text{sym}}|} \sum_{k=1}^{|O|} sd(o_k) \right)
\]  

(1)

where \( O \), \( O_{\text{sym}} \), and \( O_{\text{asym}} \) are the sets of all orbits of \( \phi \), orbits that are displayed symmetrically in \( D \), and orbits that are displayed asymmetrically in \( D \) respectively; \( o_k \), \( k = 1, 2, \ldots, |O| \) are the orbits of \( \phi \); and \( sd(o_k) \) denotes the Euclidean distance from the positions of points belonging to orbit \( o_k \) in \( D \) to exact symmetry. The second formula defines possible ranges of values for a drawing based on the number of orbits that are drawn as symmetric and is computed as:

\[
SQ_2 = \begin{cases} 
1 + \frac{|O_{\text{sym}}|}{|O|} & \text{if } |O_{\text{sym}}| = |O| \\
1 - \sum_{\phi \in O_{\text{asym}}} sd(\phi) & \text{otherwise}
\end{cases}
\]

(2)

The fraction \( 1 + \frac{|O_{\text{sym}}|}{|O|} \) limits the range of metric values based on the number of orbits displayed as symmetries in the drawing, while the sum \( \left( 1 - \sum_{\phi \in O_{\text{asym}}} sd(\phi) \right) \) penalizes the results based on the Euclidean distance from exact symmetry.

### 3.2 Symmetry Quality Metric for Automorphism Groups

Another refinement of the metric is to take as input an automorphism group, to detect the extent to which multiple automorphisms are simultaneously displayed in a drawing of a graph. Our metric is able to take as input dihedral groups, which contain both rotational and reflectional automorphisms.

To compute the group symmetry quality \( SQG \), we consider a weighted sum based on the maximum orbit size of each orbit set. Given \( A \) as the automorphism group to be checked, we define weight \( w \) as:

\[
w = \sum_{\phi \in A} K(\phi)
\]

(3)

where \( K(\phi) \) is the size of the largest orbit of an automorphism \( \phi \in A \). For reflectional automorphisms, the value of \( K(\phi) \) is always 2, while for rotational automorphisms, the value of \( K(\phi) \) is the order of the rotation. Given this weight \( w \), the group symmetry quality score \( SQG \) is computed using the formula:

\[
SQG = \begin{cases} 
\frac{1}{w} \sum_{\phi \in A} K(\phi) \times SQ(\phi) \times \frac{1}{2} & \text{if } |A_{\text{sym}}| = 0 \\
\frac{1}{w} \sum_{\phi \in A} K(\phi) \times SQ(\phi) \times \frac{1}{2} & \text{otherwise}
\end{cases}
\]

(4)

where \( A_{\text{sym}} \) is the subset of \( A \) containing automorphisms that are displayed as exact symmetries in the drawing \( D \) and \( SQ(\phi) \) is the score computed by the \( SQ \) metric with \( \phi \) as the input automorphism. \( SQ(\phi) \) can be computed with either \( SQ_1 \) (Equation 1) or \( SQ_2 \) (Equation 2).

The usage of the weight and the multiplier \( K(\phi) \) for each orbit is meant to give preference to drawings that display higher orders of symmetry, i.e. possessing more symmetries. Thus, a drawing that is symmetric but only of a lower order of symmetry will score lower than a symmetric drawing displaying a higher order of symmetry. Furthermore, adding \( 1 \) to the sum when at least an automorphism is realized as a symmetry ensures that every symmetric drawing will obtain a higher score than every asymmetric drawing.

### 4 Algorithms for the Symmetry Quality Metric

We present algorithms used to compute our metrics. Section 4.1 and 4.2 provide algorithms for detecting exact and approximate symmetries respectively within the framework of our metric. Section 4.3 presents the algorithm to compute the \( SQ \) version of our metric, and Section 4.4 presents the algorithm to compute the \( SQG \) version of our metric.

#### 4.1 Exact Symmetry Detection

While exact symmetry detection for two-dimensional point sets and polygons can be run in \( O(n \log n) \) time, node-link drawings of graphs are often not simple polygons, which necessitates a modification to the approach. With symmetric graph drawings, it is possible to detect an automorphism of the graph by detecting the geometric symmetry of the vertex point set of the drawing and checking the adjacency of the vertices. The following describes the algorithm:

**Algorithm 1: Algo-ExactSym**

**Input:** Graph \( G \), drawing \( D \) of \( G \)

**Output:** true or false

1. Sort the vertex point set of \( D \) according to their angle around and distance from the centroid.
2. Detect the geometric symmetry of the point set. If no symmetry is detected, return false, otherwise obtain the geometric symmetry \( \sigma \).
3. For each orbit \( o_k \) in \( \sigma \), compare the adjacency of the vertices included in \( o_k \). If all of the vertices do not share the same pattern of adjacency, return false.
4. Return true if for every orbit, all the vertices share the same adjacency pattern.

**Algo-ExactSym** runs in \( O(n \log n + m) \) time. Exact symmetry detection algorithms (both rotational and axial) for two-dimensional point sets, run in worst case \( O(n \log n) \) time [25]. As each vertex only appears in one orbit, the total running time of the symmetry detection step is of \( O(n \log n) \) time in the number of vertices. Step 3 takes linear time in the degree of the vertices, i.e. linear in the number of edges, putting the time complexity of the algorithm as \( O(n \log n + m) \) time, with \( n \) as the number of vertices and \( m \) as the number of edges.

Recently, De Luca et al. [20] presented a method to detect axial, rotational, and translational symmetries in graph drawings using a machine learning approach. However, this approach is a classification approach which only returns whether a drawing contains certain symmetries and with an 8% misclassification rate.

#### 4.2 Approximate Symmetry Detection

The previous approach is limited by the nature of the exact symmetry detection algorithm, which either returns a positive answer for an exact symmetry or a negative answer otherwise and does not quantify how far a negative answer is to exact symmetry. There is therefore a need for a method that is able to compare the approximate symmetry of a drawing to the automorphism of the underlying graph.

Given an algorithm that computes the approximate symmetry of a point set, we can use it to determine whether each input orbit of automorphism is displayed as a geometric symmetry in the drawing of the graph. For our implementation, we use the symmetry distance approach of Zabrodsky et al. [27] and we compute \( sd \) as mentioned in the formulas of \( SQ_1 \) and \( SQ_2 \) as below:

**Algorithm 2: Algo-ApproxSym**

**Input:** Graph \( G \), drawing \( D \) of \( G \), automorphism \( \phi \) of \( G \)

**Output:** sd

1. Compute the geometric symmetry of the vertices.
2. For each orbit, compute the symmetry distance of the points in the orbit to the geometric symmetry.
3. Take the average of the symmetry distances for each orbit.
4. Return the average as the symmetry quality score.
1. Normalize the points in the vertex point set $P$ of drawing $D$ such that all of the points lie within the unit circle centered on the centroid of the point set.

2. For each orbit $o_k$ in an automorphism $\phi$:
   
   (a) Take the subset $P_k$ of $P$, containing points corresponding to vertices contained in $o_k$.
   
   (b) Compute the symmetric configuration $P_\sigma$ closest to $P_k$.
   
   (c) For every vertex $v$ included in $o_k$, compute the Euclidean distance between its position in $D$ and its image in $P_\sigma$, average the values, and divide by 2 to obtain the average distance $d$.
   
   (d) Subtract $d$ from 1 to obtain the value $sd$.

The distance computed by the symmetry distance approach is dependent on the area taken by the drawing $D$. To normalize this distance, we scale the point set such that the whole drawing fits in a unit circle, which limits the maximum possible distance to 2. We then divide the average distance $d$ by 2 and subtract the result from 1 in order to get a score where 1 corresponds to exact symmetry and lower values correspond to drawings that are further from exact symmetry.

In theory, only orbits where the average distance $d$ is 0 should be considered symmetric. However, computers work with floating point precision numbers rather than real numbers, leading to unavoidable round-off errors. To account for this, we define a threshold $\epsilon$ such that orbits with $d$ less than $\epsilon$ are considered to be symmetric.

**Algorithm 3: Algo-SQ**

**Input**

$G$, drawing $D$ of $G$, group of automorphisms $A$ of $G$

**Output**: $SQ$ metric score

1. For each orbit $o_k$ in $A$:
   
   (a) Compute the symmetry distance-based score $sd$ using Algo-ApproxSym
   
   (b) Add $o_k$ to the set $O_{sym}$ if the value of $sd$ is less than or equal to a threshold $\epsilon$, or add it to the set $O_{asym}$ otherwise

2. Compute the symmetry quality using either $SQ_1$ or $SQ_2$.

Using Algo-ApproxSym, given a center or axis of symmetry, the $sd$ values can be computed in $O(n \log n)$ time in the number of vertices. When this information is not given, we select the center or axis of symmetry in the following way:

- For rotational symmetries, we compute the centroids for each orbit (the point itself in the case of fixed points) and select one that is closest to all other centers (i.e. geometric median).

In the case of ties, we compute the $sd$ of each orbit and select the one with the largest $sd$ score. After initially sorting the points in $O(n \log n)$ time, computing the geometric median can be done with two $O(n)$-time sweeps along the x- and y-axes and computing the $sd$ scores when needed takes $O(n)$ time, keeping the runtime at $O(n \log n)$ time.

- For axial symmetries, we compute the best line of symmetry using Singular Value Decomposition (SVD) as a pre-processing step, as proposed by Zabrodsky et al [26]. While SVD runs in $O(n^3)$ in the size of the input matrices, in our case the matrix is of a fixed size $2 \times 2$ as we only consider drawings in 2 dimensions, keeping the runtime complexity of the metric computation at $O(n \log n)$ time in the number of vertices.

Therefore, Algo-SQ runs in $O(n \log n)$ time.

**4.4 Algorithm for Computing SQG**

The $SQG$ metric takes as input multiple automorphisms of a graph that can be displayed simultaneously in the same drawing and returns a symmetry quality metric score that measures how faithfully the drawing displays all the automorphisms at once. Given a drawing $D$ of a graph $G$ and a group of automorphisms $A$ of $G$, the metric is computed using the following steps:

**Algorithm 4: Algo-SQG**

**Input**: Graph $G$, drawing $D$ of $G$, group of automorphisms $A$ of $G$

**Output**: $SQG$ metric score

1. For each automorphism $\phi$ in $A$, compute $SQ(\phi)$ using Algo-SQ

2. Compute the weighted sum of $SQ(\phi)$ for all automorphisms in $A$, where the scores are weighted by the maximum orbit size of each automorphism.


The computation of each $SQ$ score for each automorphism runs in $O(n \log n)$ time. With $k$ as the number of automorphisms in the input group, the $SQG$ computation as a whole runs in $O(kn \log n)$ time.

**5 EXPERIMENT 1: SQ METRIC VALIDATION EXPERIMENTS**

**5.1 Experiment Design**

To validate the $SQ$ version of our metric, we perform experiments where we take a symmetric drawing of a graph and deform it in a way that breaks the symmetry.

We expect that not only will our metric effectively capture the distortion from perfect symmetry induced by the deformations, but also that it will perform better than existing approximate symmetry detection approaches which only rely on Euclidean distance, here represented as $sd$. We formulate the following hypotheses:

- Hypothesis 1: The symmetry quality scores $SQ_1$ and $SQ_2$ will both decrease as the drawing $D$ of $G$ is further deformed.

- Hypothesis 2: The symmetry quality scores $SQ_1$ and $SQ_2$ will reflect the extent of distortion from perfect symmetry more effectively than $sd$.

We performed five sets of experiments each to validate our metric on rotational and axial symmetry. For each experiment, we perform the following steps:

1. For a symmetric graph $G$, select $\phi$, one of its geometric automorphisms.

2. Create a drawing $D$ of $G$ that displays $\phi$ as an exact symmetry.
3. Select a subset of vertices of $G$ and perturb their position by taking them further from their initial position. Repeat until a desired number of perturbation steps is reached.

We used a mix of well-known symmetric graphs from graph theory literature and newly-generated symmetric graphs. The graphs we generated for validation experiments with rotational symmetry (titled in the format $c[\text{order}]x[#\text{of orbits}]$) were generated as follows. First, we decide on the order of rotational symmetry $k$ and the number of orbits $m$, then define a graph $G$ with $k \times m$ vertices and initially zero edges. We then connect each of the $m$ sets of $k$ vertices into $m$ distinct cycles of length $k$, and, except for the “innermost” cycle, we connect them to previous cycles with a number of edges between $k$ and $2k$ such that the graph possesses the order of symmetry desired.

With graphs generated for axial symmetry, we first determine the orbits, including fixed points, create edges between vertices not in the same orbit, then “mirror” the edges by adding edges between the points which share orbits with the endpoint of the created edges.

5.2 Rotational Symmetry Experiments

Figure 2: Validation experiments detecting rotational symmetry (order 7) on Coxeter graph. Each subsequent perturbation destroys a new orbit or further perturbs an already destroyed one.

Figure 3: Symmetry quality metrics for the experiment in Figure 2. Both $SQ_1$ and $SQ_2$ captures the effect of the perturbations better than $sd$ which is only based on Euclidean distance, with $SQ_2$ capturing the effects of further perturbations better.

Figure 4: Validation experiment detecting axial symmetry on Heawood graph.

5.3 Axial Symmetry Experiments
We formulate the following hypotheses:

- Hypothesis 1: The symmetry quality scores $SQG_1$ and $SQG_2$ will increase as a drawing $D_i$ is more symmetric.
- Hypothesis 2: The symmetry quality scores $SQG_1$ and $SQG_2$ will decrease at a faster rate on latter perturbation steps than $SD_1$.
- Hypothesis 3: The symmetry quality scores $SQG_1$ and $SQG_2$ will decrease as a drawing $D_i$ is further deformed.
- Hypothesis 4: The symmetry quality scores $SQG_1$ and $SQG_2$ will be higher for a drawing $D_i$ that displays as symmetries a larger subset of the automorphism group or an automorphism with a larger order than a drawing $D_i$ displaying as symmetries a smaller subset of the automorphism group or an automorphism with a smaller order.

For these experiments, we start by selecting a symmetric graph $G$ and a geometric automorphism group $A$. We then create drawings of $G$ displaying different subsets of $A$ as symmetries. In addition, we also perform deformation experiments similar to that described in Section 5, where we perform steps of deformation that gradually destroys more orbits and/or brings perturbed vertices further from their initial positions. We perform six sets of experiments with drawings displaying the input automorphism groups as exact symmetries and four sets of experiments with drawing deformations.

### 6.2 Cyclic Group Experiments

Figure 5 shows an example of a validation experiment for using the $SQ$ metric with an axial automorphism as input. We use the Heawood graph in this example and draw a layout that displays one of its automorphisms as an axial symmetry, with the $y$-axis as the axis of symmetry. Each subsequent drawing in Figure 4 destroys at least one more orbit compared to the previous drawing.

Figure 5 shows the $SQ$ metrics computed for each perturbation step for the experiment depicted in Figure 4. Similar to the case seen in Figure 5, both $SQ_1$ and $SQ_2$ decrease as the perturbation steps continue and both obtain lower scores than $SD_1$ on the perturbed drawings, further supporting Hypotheses 1 and 2. Similar patterns can be seen in all other experiments, validating Hypotheses 1 and 2.

### 5.4 Discussion and Summary

In all validation experiments, $SQ_1$ and $SQ_2$ decrease steadily with the number of perturbation steps, while also capturing the changes in quality better than $SD_1$. Another point of note is that $SQ_2$ consistently decreases at a faster pace than $SQ_1$, such as can be seen in Figure 4, denoting that it can be more effective at capturing the changes in the quality of the symmetry of the drawings.

Furthermore, in cases such as in Figure 5, we see that on the later half of the perturbation steps, the scores computed by the $SQ_1$ formula do not differ much from $SD_1$, while $SQ_2$ continues to decrease to a low score of below 0.1 for step 10 where five out of seven orbits have been perturbed. This further shows that $SQ_2$ is better at capturing distortions from perfect symmetry.

In summary, our experiments have supported Hypotheses 1 and 2, validating the effectiveness of our metric to measure the quality of symmetry in graph drawings with respect to one selected graph automorphism and demonstrates its better effectiveness in measuring deviations from perfect symmetry compared to the Euclidean distance-based-only $SD_1$, as well as showing that $SQ_2$ captures the changes in quality more effectively than $SQ_1$.

### 6 Experiment 2: $SQG$ Metric Validation Experiments

#### 6.1 Experiment Design

To validate the $SQG$ version of our metric, we compare drawings that show different subsets of an automorphism group and compare the value of the metric computed on each drawing.

We expect that our metric will score a drawing higher the more automorphisms from the input automorphism group it displays as symmetries, or if it displays an automorphism with a higher order. Furthermore, as with the $SQ$ version, we expect our metric to effectively reflect the degree of approximate symmetry of a drawing and score asymmetric drawings that are closer to exact symmetry higher. We formulate the following hypotheses:
After validating the usage of the SQG metric in computing 4, validating the effectiveness of the SQG metric in capturing the difference in quality of drawings displaying different subsets of automorphisms of a graph, we conduct experiments comparing a number of different graph layout algorithms. We select the following automatic layouts to be compared: Fruchterman-Reingold (FR) [10], Stress Majorization [1], Pivot MDS [4], spectral, and Tutte. We also produce drawings using the Concentric Circles layout [1], where we create concentric circles according to the number of orbits of a selected automorphism, assign each orbit to a circle, and place the vertices belonging to the orbit in a regular convex polygon position around the circle.

We perform the experiments using the following steps:

1. We select a symmetric graph $G$ and draw it using all of the selected layout algorithms.
2. We define a group of automorphisms $A$ of $G$ as the input.
3. We compute the SQG metrics for the drawings produced by each layout algorithm with $A$ as the input automorphism group.

In Step 1, we generate one drawing each using each layout algorithm, except for FR, where we generate five drawings per graph, each time starting from a random initial layout. This is due to the non-deterministic nature of FR, compared to the other selected layout algorithms. We then average the SQG metrics computed for all the FR-generated drawings to obtain a value that can be compared to those computed for other layout algorithms.

Based on how the Concentric Circles layout places vertices in regular convex polygon positions along concentric circles, we expect that this layout will perform well in displaying automorphisms as geometric symmetry. We also expect that the Tutte layout, due to being designed to display planar graphs, will perform well in displaying planar automorphisms, which are automorphisms that can be displayed in planar graph drawings. We formulate the following hypothesis for this experiment:

Hypothesis 5: The Concentric Circle layout will always attain SQG metrics of 1 and Tutte will always display planar automorphisms when the graph has planar automorphisms.

### 7.2 Layout Comparison Results

Figure 11 shows an example of the layout comparison experiment using the dodecahedral graph. In this experiment, we use the SQG

![Figure 6: Validation experiment with rotational symmetries of orders 12, 6, 4, 3, and 2.](image)

![Figure 7: Validation experiments with dihedral symmetry of orders 10, 5, and 2 on the dodecahedral graph and dihedral symmetry groups of orders 6, 3, and 2 on the cuboctahedral graph.](image)
metric with the input dihedral groups of order 10, 5, and 2. Figure 12 displays the symmetry quality scores computed for this experiment. Concentric Circles obtains a metric score of 1, supporting Hypothesis 5. Stress Majorization also obtains a metric score of 1, and behind the two, Tutte obtains the third highest metric score. A similar result is shown in the examples using the tesseract graph in Figures 13 and 14 where we take as input dihedral groups of order 8, 4, and 2. Concentric Circles and Stress Majorization again obtain SQG values of 1, with Tutte the third highest.

Figure 15 shows another example of the layout comparison experiment, taking as input dihedral group of order 5 on the Petersen graph. Figure 16 shows the SQG1 and SQG2 computed for this experiment. Although the automorphism is not planar, Tutte obtains SQG of one, the same as Concentric Circles, while all other layouts obtain SQG of lower than 0.3.

7.3 Discussion and Summary

The results for layout comparisons, as summarized in the averages over nine graphs shown in Figure 10, show that the Concentric Circles, Tutte, and Stress Majorization layouts obtain the top 3 values for the SQG metric. The results for Concentric Circles support Hypothesis 5.

The results for the Tutte layout may arise from the nature of the layout algorithm, where given a set of vertices forming the outer face fixed as a regular convex polygon, all other vertices are placed at the barycenter of its neighbors. This results in a unique embedding that minimizes a linear system of equations, which could lead to the ability to capture automorphisms of the graph as symmetries. However, as the layout is designed for planar graphs, the results favor planar automorphisms. This causes it to score lower on examples such as the dodecahedral graph in Figure 11 where it displays a dihedral automorphism of order 5, compared to Stress Majorization which displays the (non-planar) dihedral automorphism of order 10.

With Stress Majorization, it has been shown that it is possible to obtain symmetric layouts with low stress starting from the Concentric Circles layout [9]. Unlike force-directed layouts such as FR, the set of minima for stress-based layouts is more limited, increasing the chances of producing a symmetric layout, which can explain its superior performance compared to FR.

In summary, our experiments have supported hypothesis 5 by showing that the concentric circles layout always obtains a score of 1 on SQG and that Tutte layout is always able to display the highest order of planar automorphism when a graph possesses planar automorphisms. We also observe that Stress Majorization performs better at displaying graph automorphisms as symmetries compared to the remaining tested layouts: FR, Pivot MDS, and Spectral.

8 Conclusion

We have introduced a new quality metric to measure how faithfully a graph drawing visualizes the graph’s automorphisms as symmetries, based on both Euclidean distance and mathematical group theory. The metrics, SQ and SQG, are suited to detect symmetries corresponding to rotational, reflectional, and dihedral automorphisms.

We defined algorithms to compute the metrics, running in O(n log n) time for the SQ version taking as input a single automorphism and in O(kn log n) time for the SQG version taking as input an automorphism group, with n as the number of vertices and k as the size of the automorphism group.

Experiments have validated the effectiveness of both SQ and SQG in capturing the extent to which the drawing’s symmetries reflect the automorphisms of the graph. We demonstrated the effectiveness of SQ in reflecting the extent of distortion of a drawing from exact symmetry, and the effectiveness of SQG in reflecting the difference in quality between drawings displaying different automorphism groups of a graph.

We have also compared layout algorithms using our metric, and in the process, confirm the effectiveness of the concentric circles layout in displaying a graph’s automorphisms as symmetries and similarly for the Tutte layout for planar automorphisms.

Future work may include extending the metric to graph drawings in three dimensions.
Figure 10: Average $SQG$ metrics computed for all nine layout comparison graphs. Concentric Circles, Tutte, and Stress Majorization obtain the three highest scores.

Figure 11: Layout comparison experiment with the dodecahedral graph, detecting dihedral symmetries of order 10, 5, and 2. Among automatic layouts, Stress Majorization realizes all of the input automorphisms, while Tutte displays the largest planar automorphism as dihedral symmetry of order 5.

Figure 12: $SQG$ metrics computed for the layout comparison experiment with the dodecahedral graph (Figure 11). Concentric Circles and Stress Majorization obtain scores of 1, while Tutte, which displays a planar automorphism of a lower order, obtains lower scores than the two.

Figure 13: Layout comparison experiment with the tesseract graph, detecting dihedral symmetries of order 8, 4, and 2. Similar to the dodecahedral graph in Figure 11, Stress Majorization displays the highest order of automorphism (order 8) while Tutte realizes the highest order of planar automorphism with dihedral symmetry of order 4.
Figure 14: SQG metrics computed for the layout comparison experiment with the tesseract graph (Figure 13). Concentric Circles and Stress Majorization, displaying the highest order of automorphism, obtains scores of 1.

Figure 15: Layout comparison experiment with the Petersen graph, with input dihedral group of order 5. Aside from concentric circles, Tutte is the only layout that perfectly displays the automorphism group as symmetries.

Figure 16: SQG metrics computed for the layout comparison experiment with the Petersen graph (Figure 15). Concentric Circles and Tutte obtain scores of 1, while all other layouts obtain scores of less than 0.3.

REFERENCES


