

gdMetriX

A networkX Extension for Graph Drawing Metrics

Sebastian Röder · Martin Nöllenburg · Markus Wallinger

21.05.2023

TU WIEN ac ALGORITHMS AND COMPLEXITY GROUP

networkX

- Well established Python package for...

  "the creation, manipulation, and study of the structure, dynamics, and functions of complex networks"



NetworkX
Network Analysis in Python

networkX

- Well established Python package for...

  "the creation, manipulation, and study of the structure, dynamics, and functions of complex networks"



NetworkX
Network Analysis in Python

- 5.2K dependent packages, 3.14K forks
- > 52 million downloads in last 30 days

networkX

- Well established Python package for...

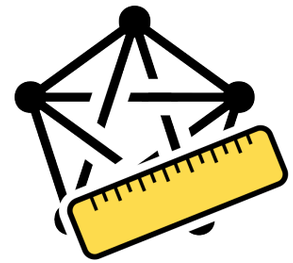
  "the creation, manipulation, and study of the structure, dynamics, and functions of complex networks"



NetworkX
Network Analysis in Python

- 5.2K dependent packages, 3.14K forks
- > 52 million downloads in last 30 days

- Goal: Extension for graph drawing metrics



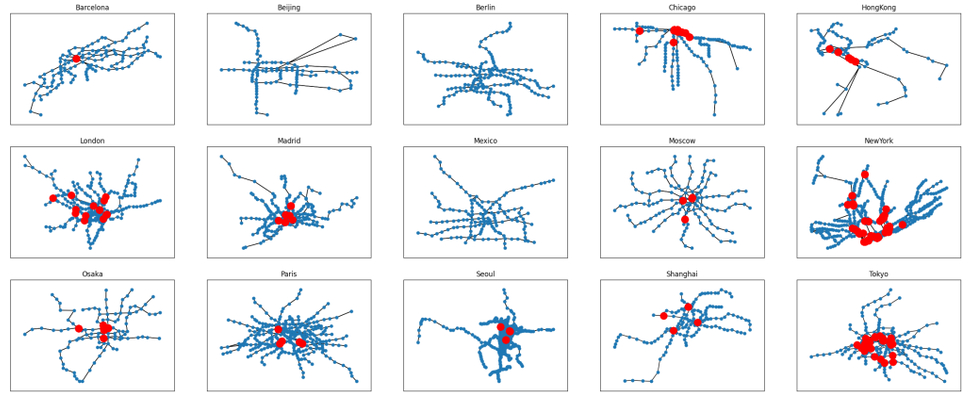
gdMetriX

Features

- Loading standard benchmark datasets

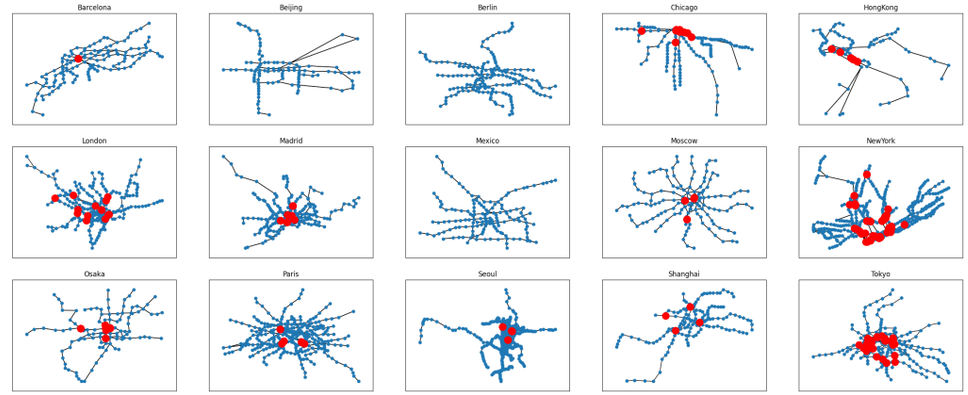
Features

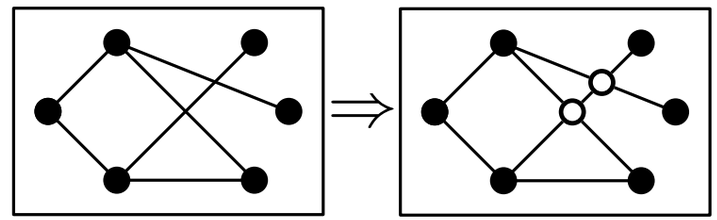
- Loading standard benchmark datasets
- 23 graph drawing metrics
  - Crossings
  - Orthogonality
  - Symmetry
  - ...



Features

- Loading standard benchmark datasets
- 23 graph drawing metrics
  - Crossings
  - Orthogonality
  - Symmetry
  - ...
- Embedding manipulation
  - Planarization
  - Combinatorial embedding

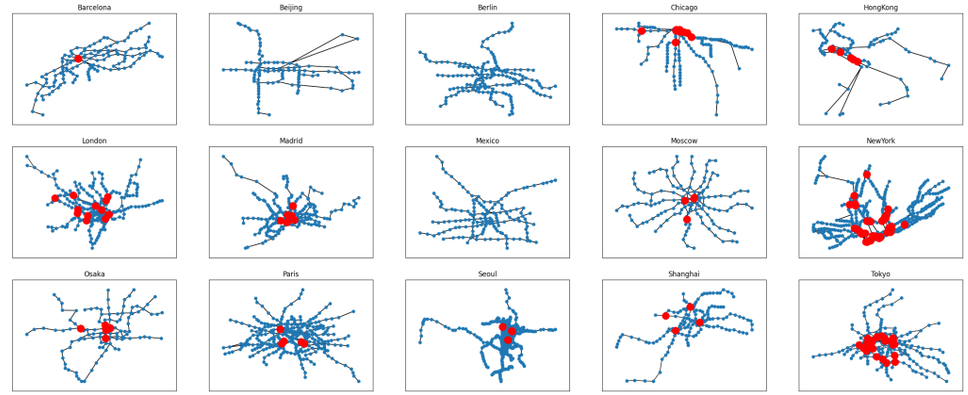


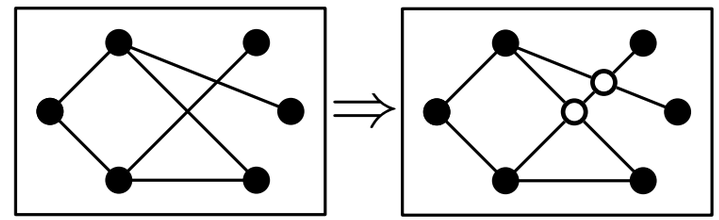


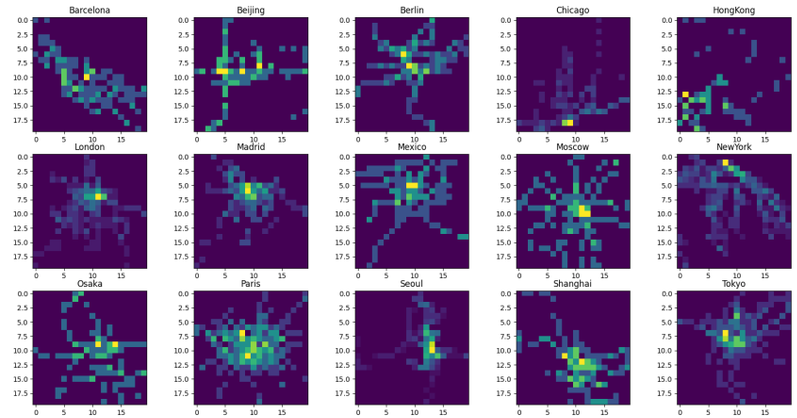
Features

- Loading standard benchmark datasets
- 23 graph drawing metrics
  - Crossings
  - Orthogonality
  - Symmetry
  - ...
- Embedding manipulation
  - Planarization
  - Combinatorial embedding
- Heatmaps

  Edge/node/positional properties $\Rightarrow$ heatmap





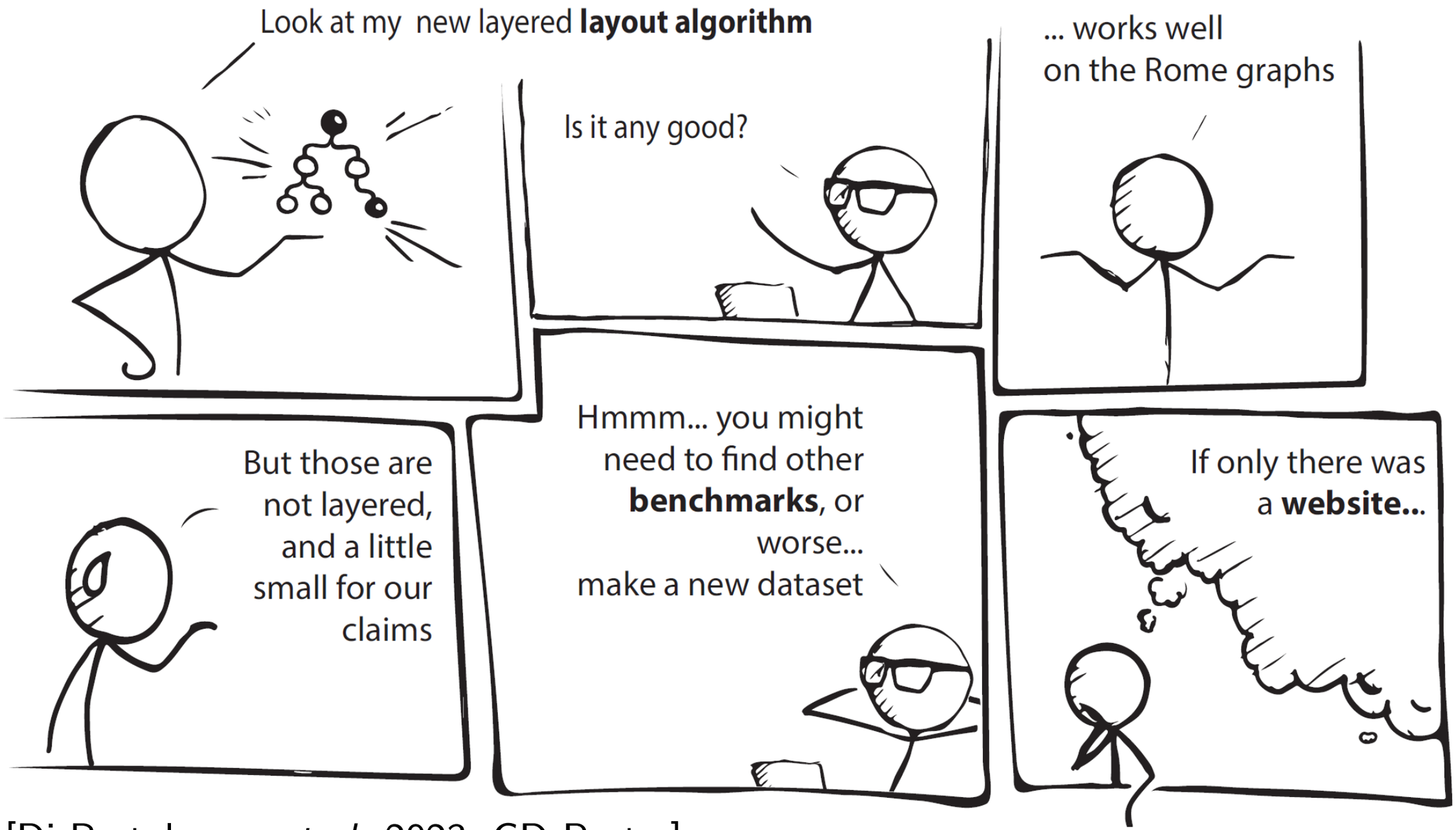


Features

- Loading standard benchmark datasets
- 23 graph drawing metrics
  - Crossings
  - Orthogonality
  - Symmetry
  - ...
- Embedding manipulation
  - Planarization
  - Combinatorial embedding
- Heatmaps

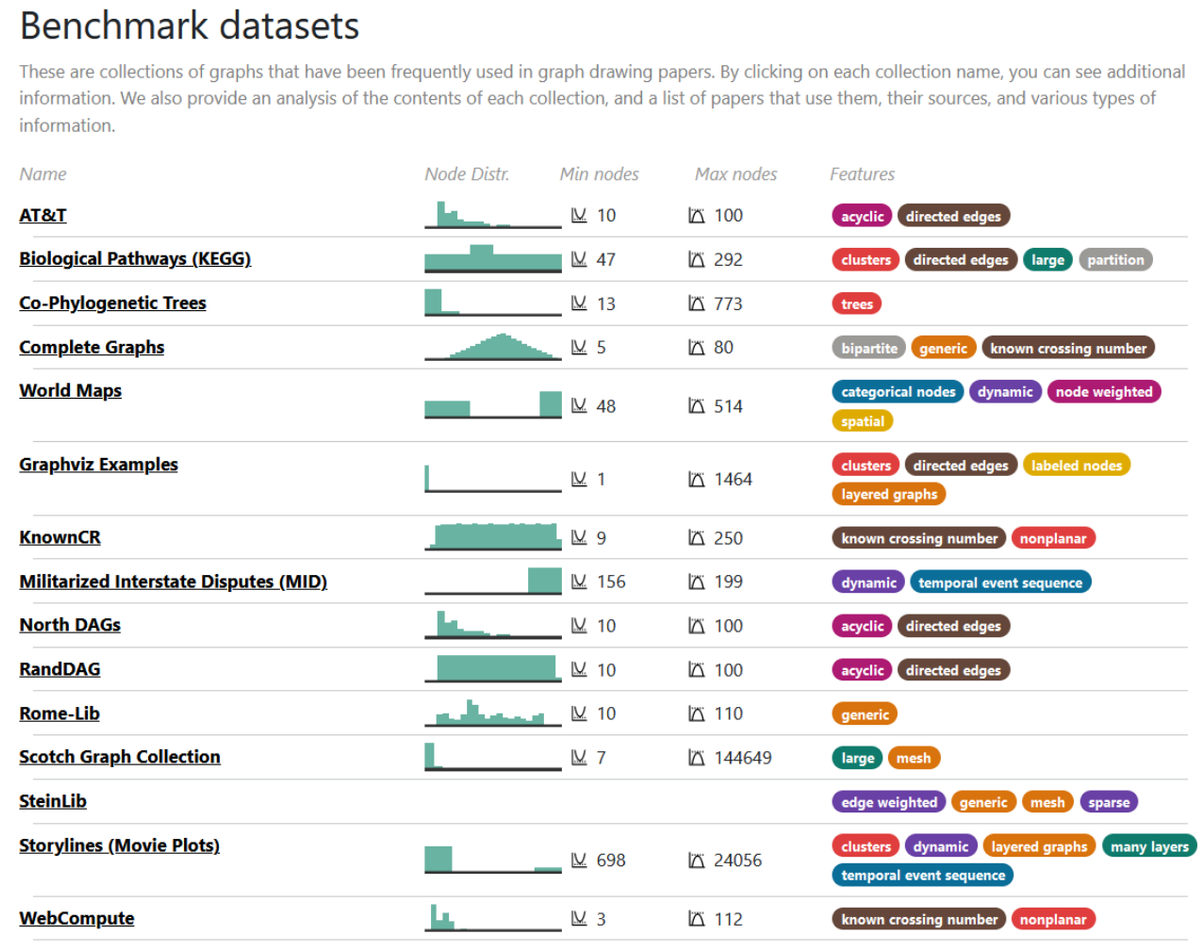
  Edge/node/positional properties $\Rightarrow$ heatmap

Datasets



[Di Bartolomeo *et al.* 2023, GD Poster]

Benchmark datasets



Benchmark datasets

These are collections of graphs that have been frequently used in graph drawing papers. By clicking on each collection name, you can see additional information. We also provide an analysis of the contents of each collection, and a list of papers that use them, their sources, and various types of information.

Name	Node Distr.	Min nodes	Max nodes	Features
AT&T		10	100	acyclic, directed edges
Biological Pathways (KEGG)		47	292	clusters, directed edges, large, partition
Co-Phylogenetic Trees		13	773	trees
Complete Graphs		5	80	bipartite, generic, known crossing number
World Maps		48	514	categorical nodes, dynamic, node weighted, spatial
Graphviz Examples		1	1464	clusters, directed edges, labeled nodes, layered graphs
KnownCR		9	250	known crossing number, nonplanar
Militarized Interstate Disputes (MID)		156	199	dynamic, temporal event sequence
North DAGs		10	100	acyclic, directed edges
RandDAG		10	100	acyclic, directed edges
Rome-Lib		10	110	generic
Scotch Graph Collection		7	144649	large, mesh
SteinLib				edge weighted, generic, mesh, sparse
Storylines (Movie Plots)		698	24056	clusters, dynamic, layered graphs, many layers, temporal event sequence
WebCompute		3	112	known crossing number, nonplanar

[Di Bartolomeo et al., Northeastern University Visulization Lab, 2023]

Benchmark datasets

- Published on **OSF**

Benchmark datasets

These are collections of graphs that have been frequently used in graph drawing papers. By clicking on each collection name, you can see additional information. We also provide an analysis of the contents of each collection, and a list of papers that use them, their sources, and various types of information.

Name	*Node Distr.*	*Min nodes*	*Max nodes*	*Features*
AT&T		10	100	acyclic, directed edges
Biological Pathways (KEGG)		47	292	clusters, directed edges, large, partition
Co-Phylogenetic Trees		13	773	trees
Complete Graphs		5	80	bipartite, generic, known crossing number
World Maps		48	514	categorical nodes, dynamic, node weighted, spatial
Graphviz Examples		1	1464	clusters, directed edges, labeled nodes, layered graphs
KnownCR		9	250	known crossing number, nonplanar
Militarized Interstate Disputes (MID)		156	199	dynamic, temporal event sequence
North DAGs		10	100	acyclic, directed edges
RandDAG		10	100	acyclic, directed edges
Rome-Lib		10	110	generic
Scotch Graph Collection		7	144649	large, mesh
SteinLib				edge weighted, generic, mesh, sparse
Storylines (Movie Plots)		698	24056	clusters, dynamic, layered graphs, many layers, temporal event sequence
WebCompute		3	112	known crossing number, nonplanar

[Di Bartolomeo et al., Northeastern University Visulization Lab, 2023]

Benchmark datasets

- Published on **OSF**
- 26 datasets, 19.597 graphs, 1.7 GB

Benchmark datasets

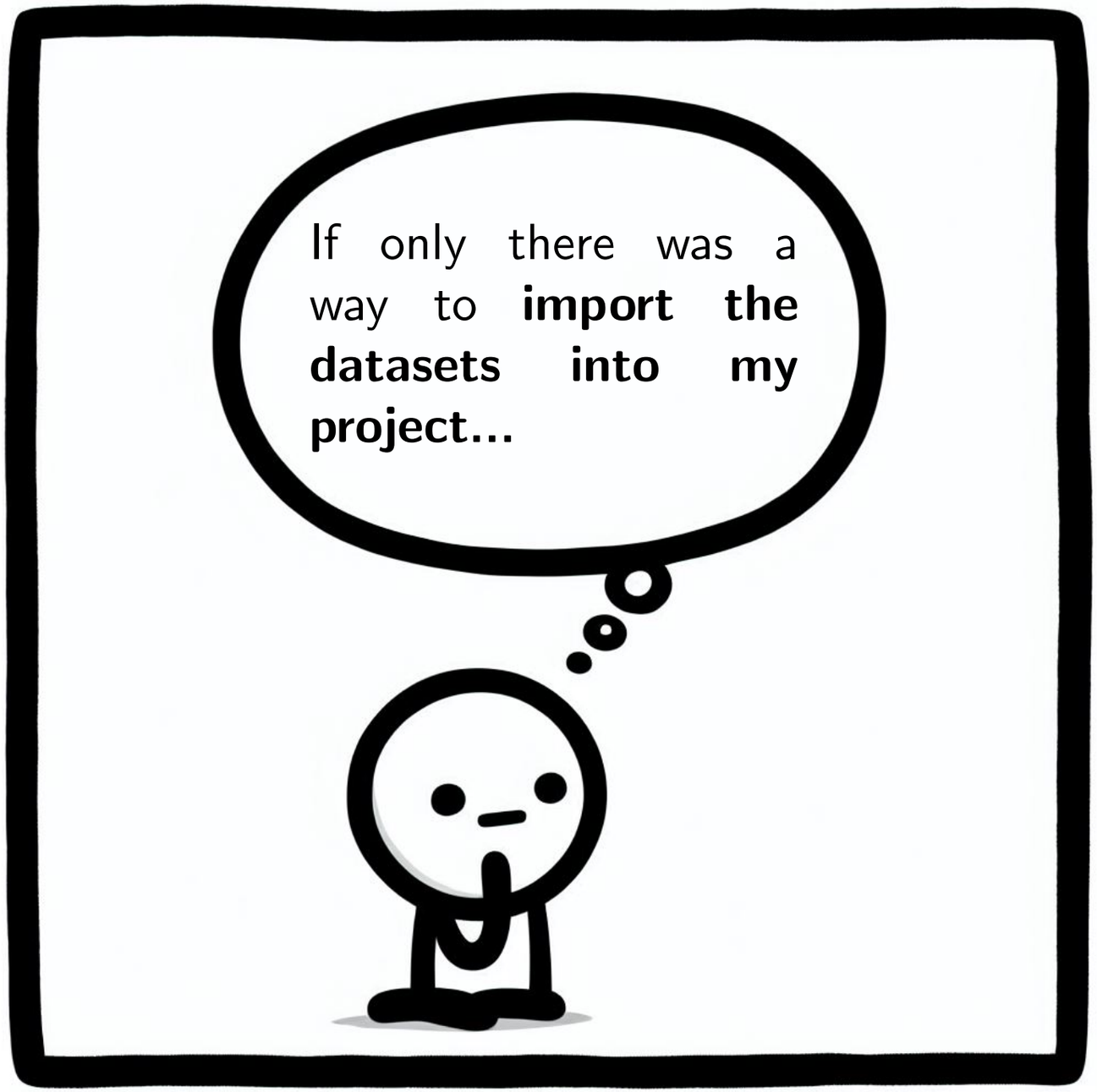
These are collections of graphs that have been frequently used in graph drawing papers. By clicking on each collection name, you can see additional information. We also provide an analysis of the contents of each collection, and a list of papers that use them, their sources, and various types of information.

Name	Node Distr.	Min nodes	Max nodes	Features
AT&T		10	100	acyclic, directed edges
Biological Pathways (KEGG)		47	292	clusters, directed edges, large, partition
Co-Phylogenetic Trees		13	773	trees
Complete Graphs		5	80	bipartite, generic, known crossing number
World Maps		48	514	categorical nodes, dynamic, node weighted, spatial
Graphviz Examples		1	1464	clusters, directed edges, labeled nodes, layered graphs
KnownCR		9	250	known crossing number, nonplanar
Militarized Interstate Disputes (MID)		156	199	dynamic, temporal event sequence
North DAGs		10	100	acyclic, directed edges
RandDAG		10	100	acyclic, directed edges
Rome-Lib		10	110	generic
Scotch Graph Collection		7	144649	large, mesh
SteinLib				edge weighted, generic, mesh, sparse
Storylines (Movie Plots)		698	24056	clusters, dynamic, layered graphs, many layers, temporal event sequence
WebCompute		3	112	known crossing number, nonplanar

[Di Bartolomeo et al., Northeastern University Visulization Lab, 2023]

Benchmark datasets

- Published
- 26 dataset
  1.7 GB



apers. By clicking on each collection name, you can see additional list of papers that use them, their sources, and various types of

Max nodes	Features
100	acyclic, directed edges
292	clusters, directed edges, large, partition
773	trees
80	bipartite, generic, known crossing number
514	categorical nodes, dynamic, node weighted, spatial
1464	clusters, directed edges, labeled nodes, layered graphs
250	known crossing number, nonplanar
199	dynamic, temporal event sequence
100	acyclic, directed edges
100	acyclic, directed edges
110	generic
144649	large, mesh
	edge weighted, generic, mesh, sparse
24056	clusters, dynamic, layered graphs, many layers, temporal event sequence
112	known crossing number, nonplanar

theastern University Vi-

Benchmark datasets

- Well, there is...

Benchmark datasets

- Well, there is...
- Get available datasets

```
>>> available_datasets = get_available_datasets()
>>> print(available_datasets)
['subways', 'code', 'rome', 'chess', 'steinlib', ...
```

Benchmark datasets

- Well, there is...
- Get available datasets

```
>>> available_datasets = get_available_datasets()
>>> print(available_datasets)
['subways', 'code', 'rome', 'chess', 'steinlib', ...
```

- Iterate all graphs of a dataset

```
>>> for name, g in iterate_dataset('subways'):
>>>     print("'{name}' has {n} vertices and {m} edges".format(name=name, n=g.order(), m=le
```

Benchmark datasets

- Well, there is...
- Get available datasets

```
>>> available_datasets = get_available_datasets()
>>> print(available_datasets)
['subways', 'code', 'rome', 'chess', 'steinlib', ...
```

- Iterate all graphs of a dataset

```
>>> for name, g in iterate_dataset('subways'):
>>>     print("'{name}' has {n} vertices and {m} edges".format(name=name, n=g.order(), m=le
```

- Automatic downloading, caching and loading

Metrics

Journal of Visual Languages and Computing (2002) **13,** 501–516
doi:10.1006/S1045-926X(02)00016-2, available online at http://www.idealibrary.com on IDEAL®

Metrics for Graph Drawing Aesthetics

HELEN C. PURCHASE

School of Computer Science and Electrical Engineering, The University of Queensland, St. Lucia 4072, Queensland, Australia

The Multi-Dimensional Landscape of Graph Drawing Metrics

Gavin J. Mooney, Helen C. Purchase, Michael Wybrow, and Stephen G. Kobourov

Abstract—Any graph drawing can be characterised by a range of computational aesthetic metrics. For example, a given drawing might be described as having eight crossings, a mean angular resolution of 0.34, and an edge orthogonality value of 0.72. However, without knowing the distribution of these metrics it is hard to compare the quality of drawings of different graphs, nor know whether a given drawing is typical or an outlier within the space of all possible drawings. This paper explores the range and distribution of ten normalised graph drawing layout metrics, based on graphs created by six graph generation algorithms and drawings created by six popular layout algorithms. We include the "Rome" and "North" graph repositories in our analysis. Our exploration of the multidimensional aesthetics space allows for comparisons between the graph drawing algorithms, highlighting those that cover larger or smaller volumes of the aesthetics space. We calculate the correlation coefficients between the metrics, indicating those that may conflict with each other (negatively correlated), and those that may be redundant (positively correlated). Our results will be useful as the basis for simulated annealing or gradient descent layout algorithms, for identifying the best layout algorithms for producing a specified combination and range of aesthetics, and for informing experimental controls in human empirical studies.

Index Terms—Graph layout aesthetics, Graph metrics, Graph layout algorithms

Applying Graphical Design Techniques to Graph Visualisation

Martyn Taylor, Peter Rodgers
University of Kent, University of Kent
{mt63@kent.ac.uk, P.J.Rodgers@kent.ac.uk}

Abstract

This paper contains details of new criteria for graph layout, based on the concepts used in graphical design. Current graph layout criteria have been shown to be effective in measuring the quality of a graph layout, but they are ad-hoc and often miss subtle appearance considerations such as balance and distribution.

We discuss how the principles concerning the layout of text and diagrams from design can be applied to graph layout and show how two new metrics can be implemented based on these criteria. We also give preliminary examples of layout generated using the new metrics.

1. Introduction

Over the past few years there has been much work in the field of graph visualisation with the goal of improve the layout of graphs to increase users' comprehension of the information represented by the diagram. A widely applied technique in this field is to measure the quality of a layout using a number metrics. The result of these can

2.2. In both fields there has been shown to be a positive correlation between the criteria when applied to displayed material and usability [2] [7] [8].

In order to adapt graphic design criteria for use in graph layout, it is necessary to develop measurement methods for them. We illustrate this with metrics that attempt to measure two criteria, Concentration and Homogeneity. Concentration is how evenly distributed a graph is within the area it occupies. Homogeneity is how even the distribution of graph elements is between top and bottom, or left and right.

The rest of the paper is organised as follows. The largest section, Section 3 describes implementations for both the more widely used graph drawing criteria and two criteria, Concentration and Homogeneity, which have been used in graphic design. Section 3.3 illustrates the use of the criteria in a multi-criteria graph drawing system. Section 4 gives our conclusions and details some further work to move on from this preliminary investigation.

2. Design Criteria

In this section we consider a number of criteria for producing good layout. First, we briefly summarise the

Computational Aesthetics in Graphics, Visualization, and Imaging (2007)
D. W. Cunningham, G. Meyer, L. Neumann (Editors)

The Aesthetics of Graph Visualization

Chris Bennett, Jody Ryall, Leo Spalteholz and Amy Gooch[1]

[1]University of Victoria, Victoria, BC, Canada

Abstract

The discipline of graph visualization produces pictorial representations of node–link structures. Much effort has been directed toward making such diagrams visually pleasing. A variety of aesthetic heuristics have been proposed, with the assumption that these will improve readability and understanding. We look at a perceptual basis for these heuristics, including Gestalt principles and Norman's emotional design framework. Next, we review the work to date on aesthetic heuristics and examine what has been done to evaluate these heuristics. We summarize this in a framework that outlines graph drawing heuristics, their perceptual basis, and evaluation status.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Line and Curve Generation

1. Introduction

Whether visualizations are designed to convey an existing idea or to support the formulation of new ideas, the key question remains "what makes a good visualization?" [HBW05]. In the field of graph drawing, a good visualization emphasizes readability and promotes understanding [Fis06]. Research in this area has led to aesthetic heuristics for drawing graphs. Recently, these efforts have expanded to empirically evaluate the effectiveness of these heuristics and to examine

and preventing edges from becoming long and convoluted. Semantic issues are concerned with highlighting the important characteristics of the underlying model and are influenced by the task to be performed with this data. While syntactic issues have received the most attention, there has been some research into domain specific issues in the areas of software engineering [PCA02] and social networks [HHE05, MBK97].

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Boundary
Area
Aspect ratio

Crossings
Crossings
Crossing angles

Symmetry
Node-based
Edge-based
Visual

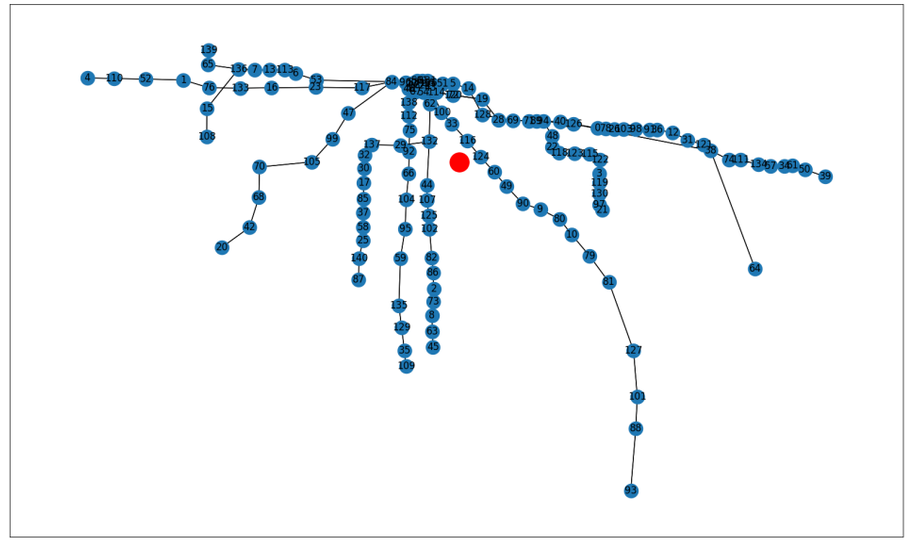
Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Center of mass

Average vertex position



Optionally weighted

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Closest pair of points

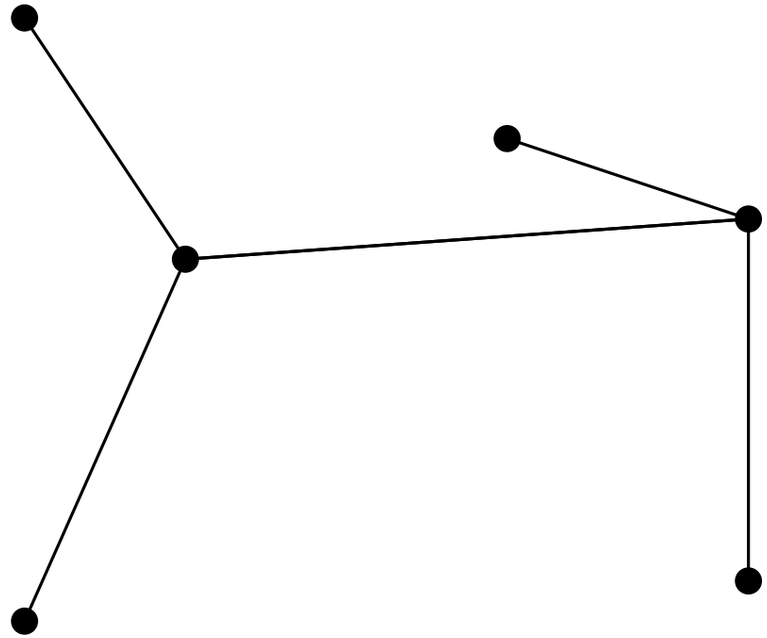
Simple $O(n \log n)$ divide-and-conquer approach

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Closest pair of elements

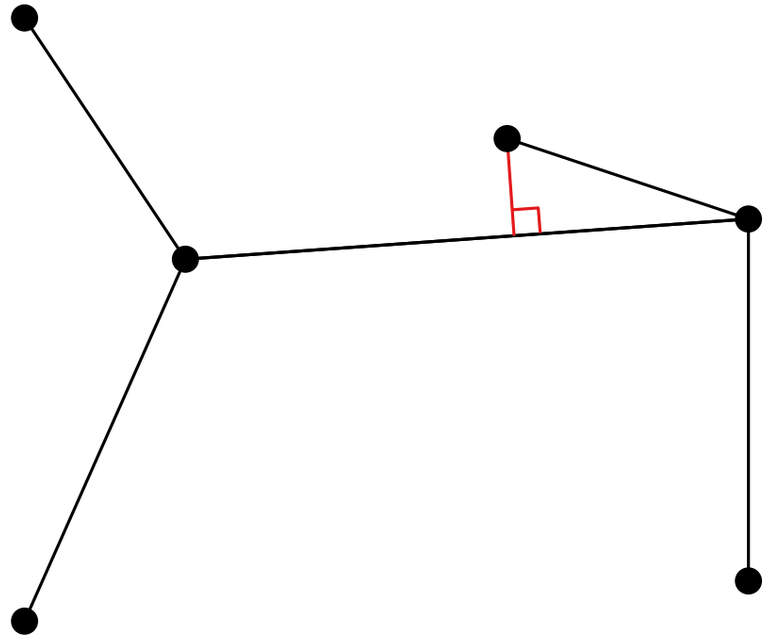


Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Closest pair of elements

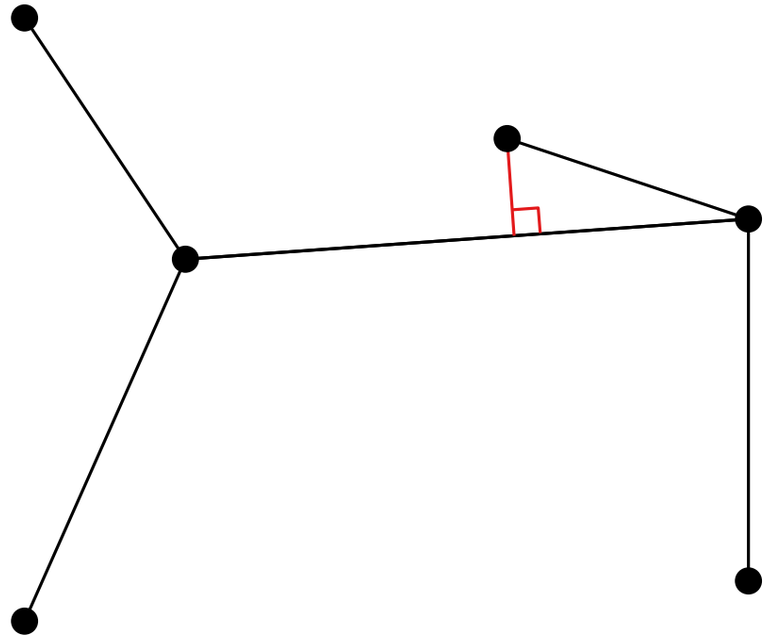


Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Closest pair of elements



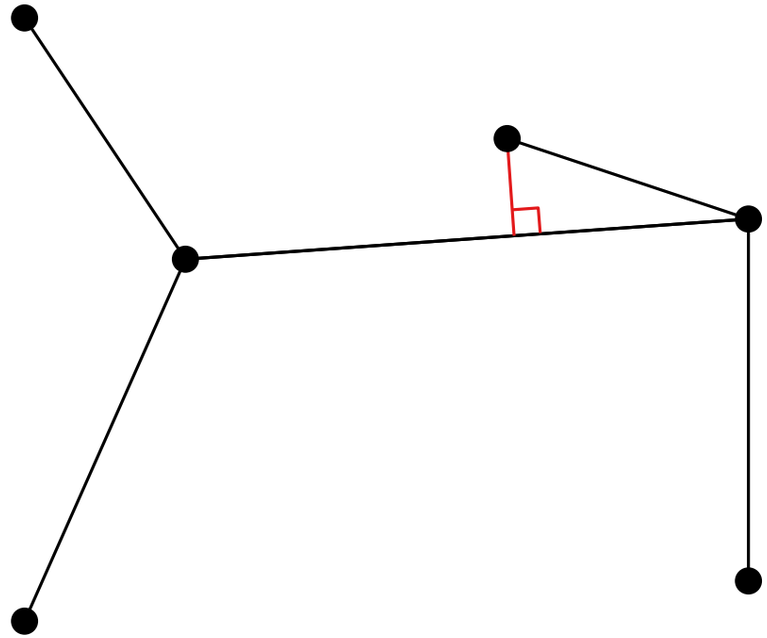
Node $\leftrightarrow$ Node
Node $\leftrightarrow$ Edge

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Closest pair of elements



Node $\leftrightarrow$ Node
Node $\leftrightarrow$ Edge

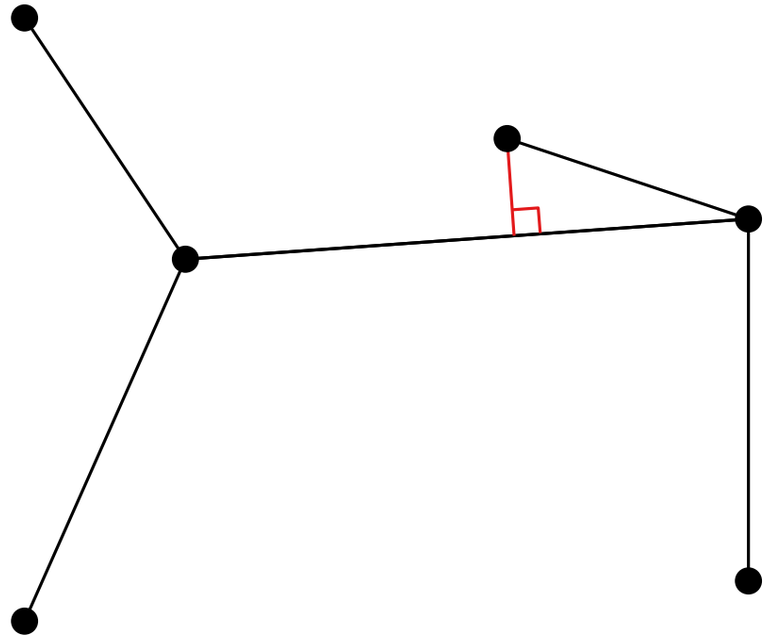
Edge $\leftrightarrow$ Edge?

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

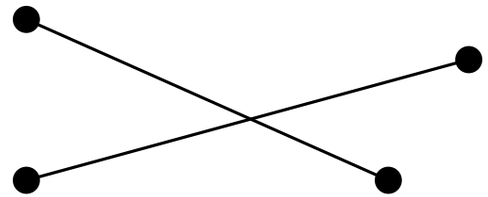
Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Closest pair of elements



Node $\leftrightarrow$ Node
Node $\leftrightarrow$ Edge

Edge $\leftrightarrow$ Edge?

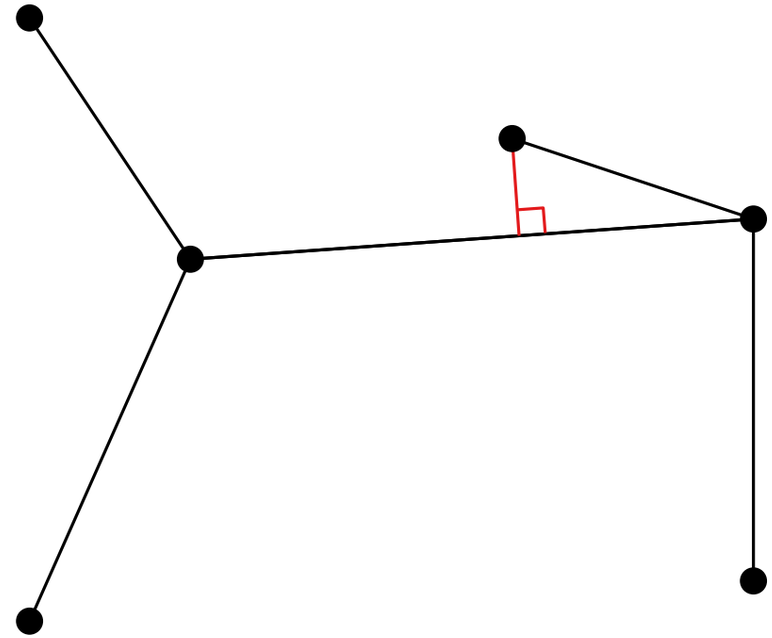


Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

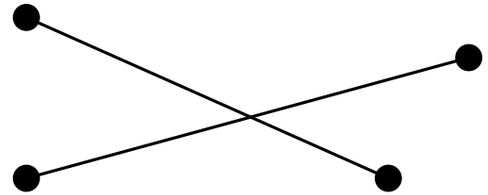
Closest pair of elements



Optional

Node $\leftrightarrow$ Node
Node $\leftrightarrow$ Edge

Edge $\leftrightarrow$ Edge?

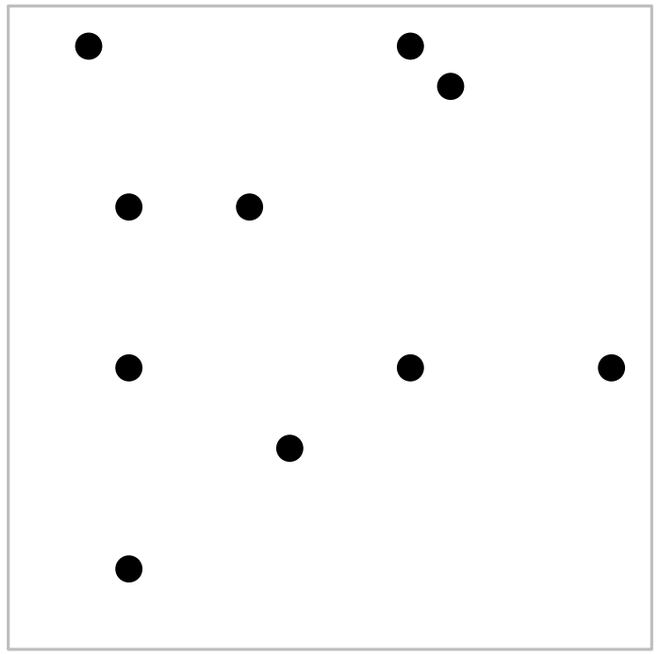


Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Concentration [Taylor, Rodgers, 2005]

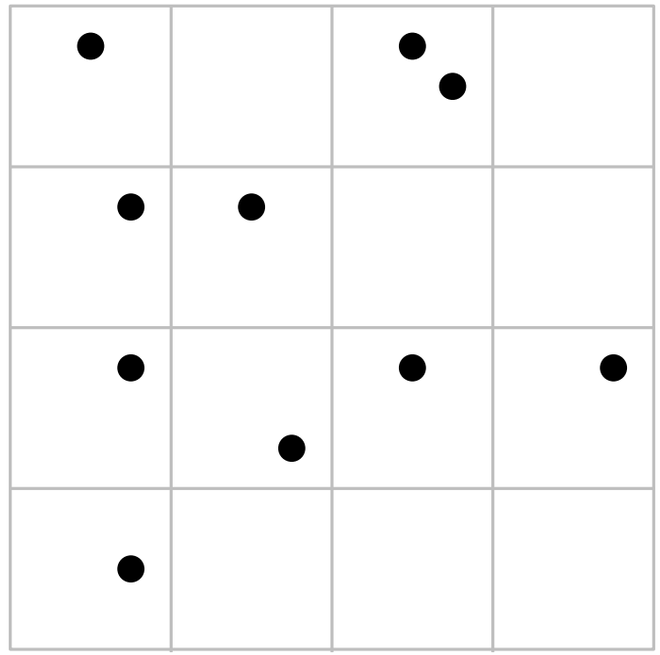


Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Concentration [Taylor, Rodgers, 2005]



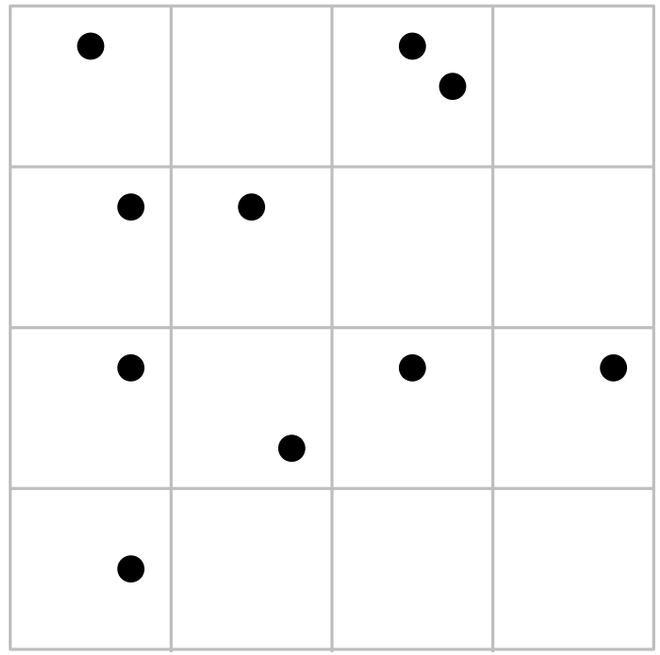
$\lceil\sqrt{n}\rceil \times \lceil\sqrt{n}\rceil$-grid

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Concentration [Taylor, Rodgers, 2005]



$\lceil\sqrt{n}\rceil \times \lceil\sqrt{n}\rceil$-grid

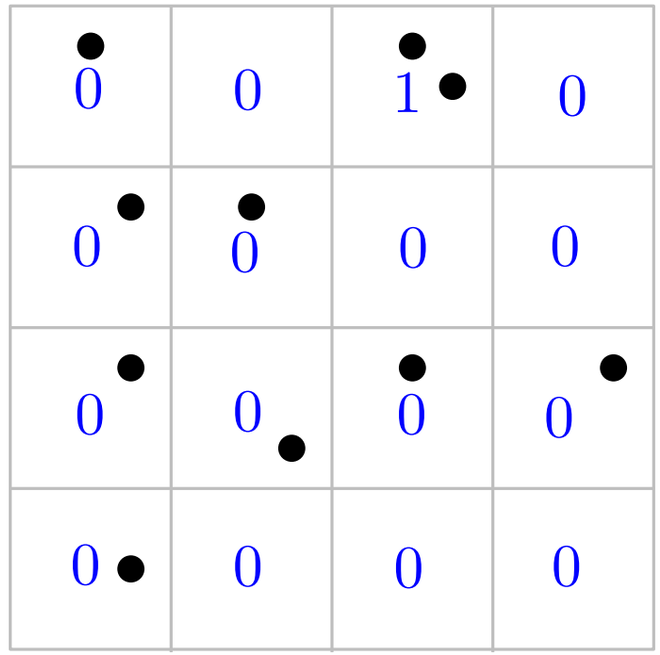
$$1 - \frac{\sum_{\text{grid cell } g_i} \max(\text{nodes in } g_i - 1, 0)}{n-1}$$

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Concentration [Taylor, Rodgers, 2005]



$\lceil\sqrt{n}\rceil \times \lceil\sqrt{n}\rceil$-grid

$n = 10$
$\lceil\sqrt{n}\rceil = 4$

$1 - \frac{1}{10-1} = 0.888$

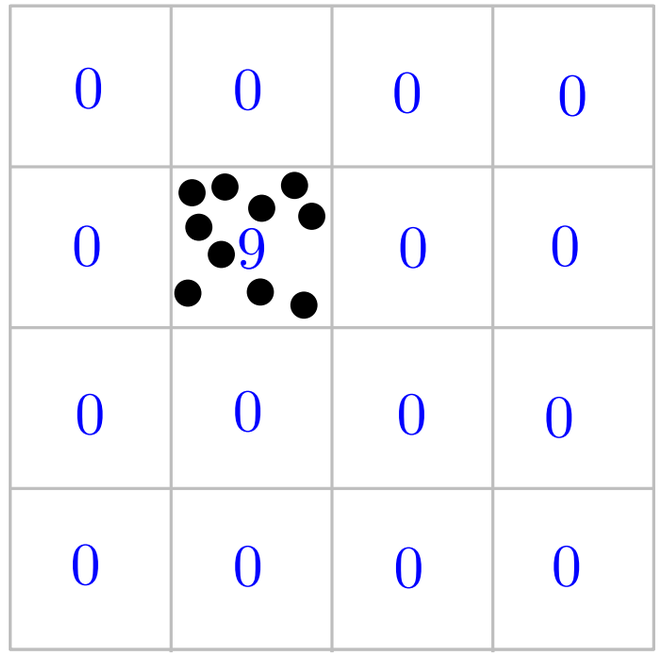
$$1 - \frac{\sum_{\text{grid cell } g_i} \max(\text{nodes in } g_i - 1, 0)}{n-1}$$

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Concentration [Taylor, Rodgers, 2005]



$\lceil\sqrt{n}\rceil \times \lceil\sqrt{n}\rceil$-grid

$n = 10$
$\lceil\sqrt{n}\rceil = 4$

$1 - \frac{9}{10-1} = 0$

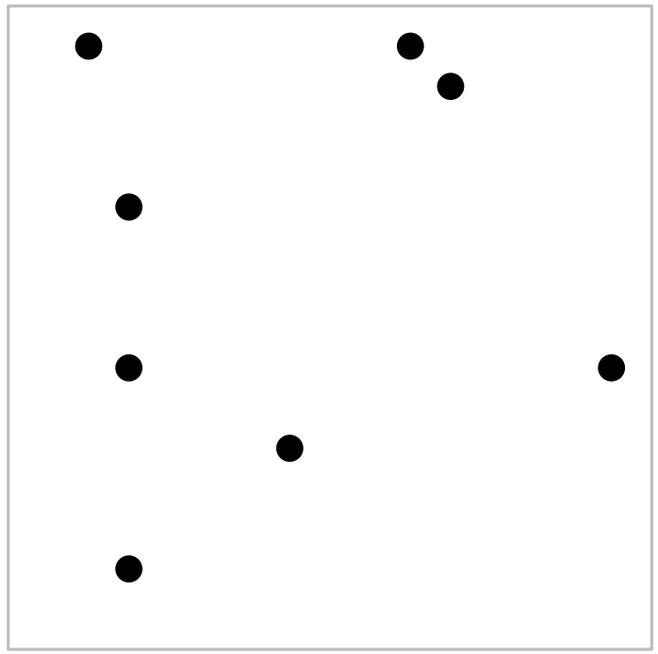
$$1 - \frac{\sum_{\text{grid cell } g_i} \max(\text{nodes in } g_i - 1, 0)}{n-1}$$

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
~~Homogeneity~~
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Homogeneity [Taylor, Rodgers, 2005]

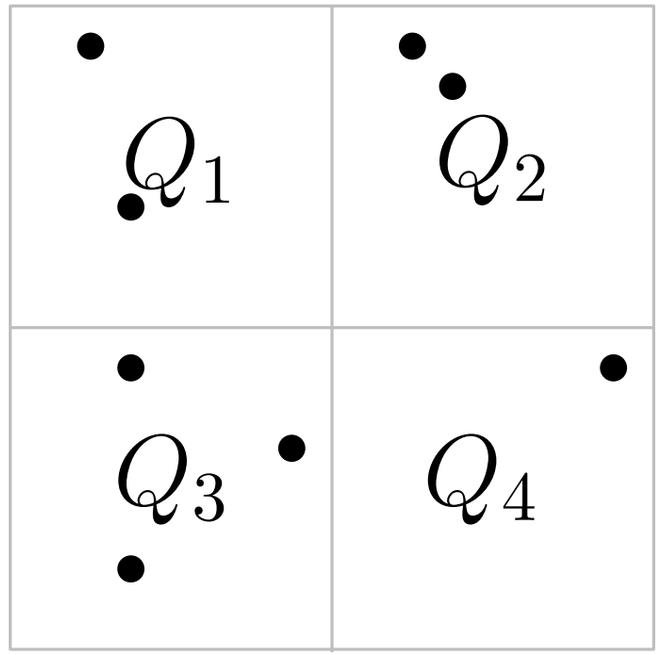


Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
~~Homogeneity~~
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Homogeneity [Taylor, Rodgers, 2005]

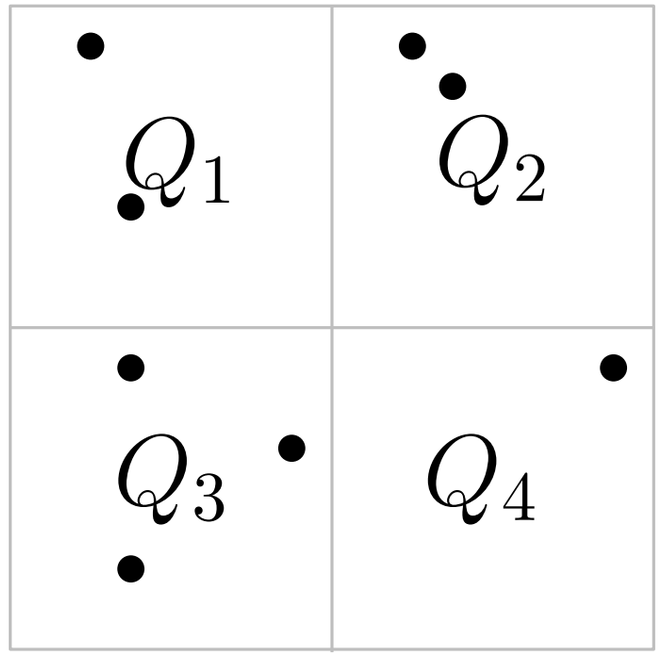


Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
~~Homogeneity~~
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Homogeneity [Taylor, Rodgers, 2005]



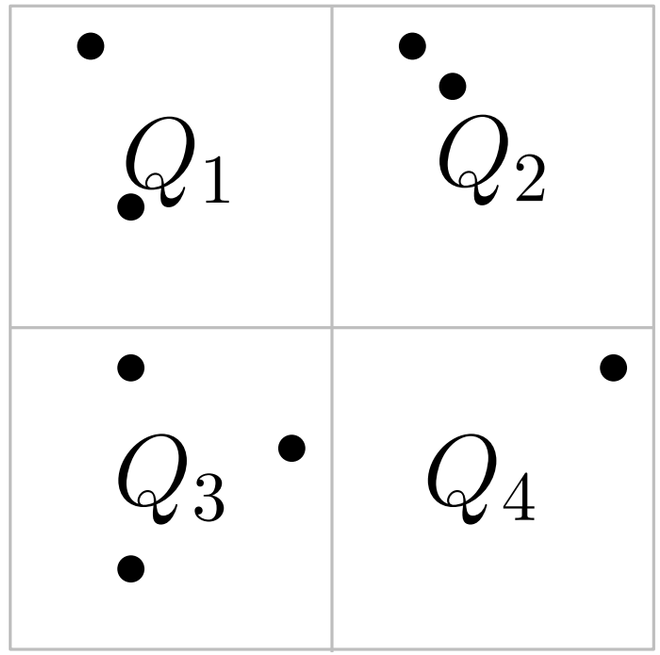
$$\frac{\lfloor \frac{n}{4} \rfloor !^4}{n_{Q_1}! n_{Q_2}! n_{Q_3}! n_{Q_4}!}$$

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
~~Homogeneity~~
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Homogeneity [Taylor, Rodgers, 2005]



$$\frac{2!^4}{2!\cdot 2!\cdot 3!\cdot 1!} = \frac{2}{3}$$

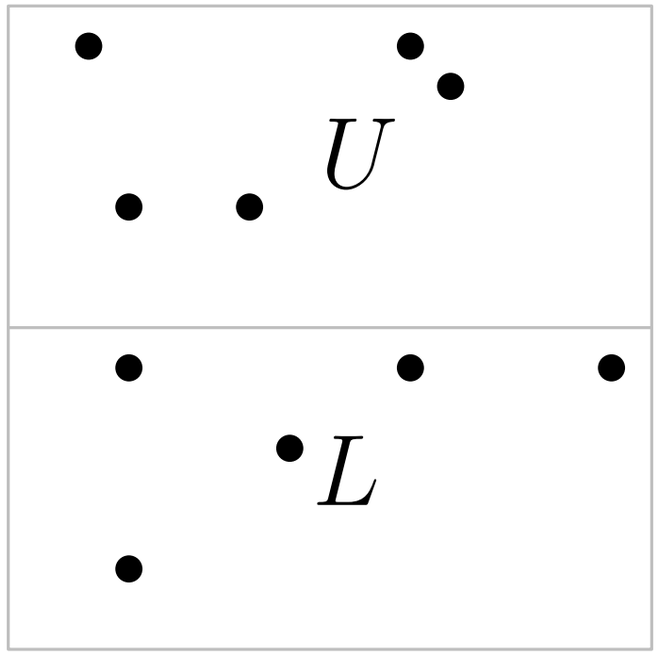
$$\frac{\lfloor \frac{n}{4} \rfloor !^4}{n_{Q_1}! n_{Q_2}! n_{Q_3}! n_{Q_4}!}$$

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Balance [Tamassia *et al.* 1988]



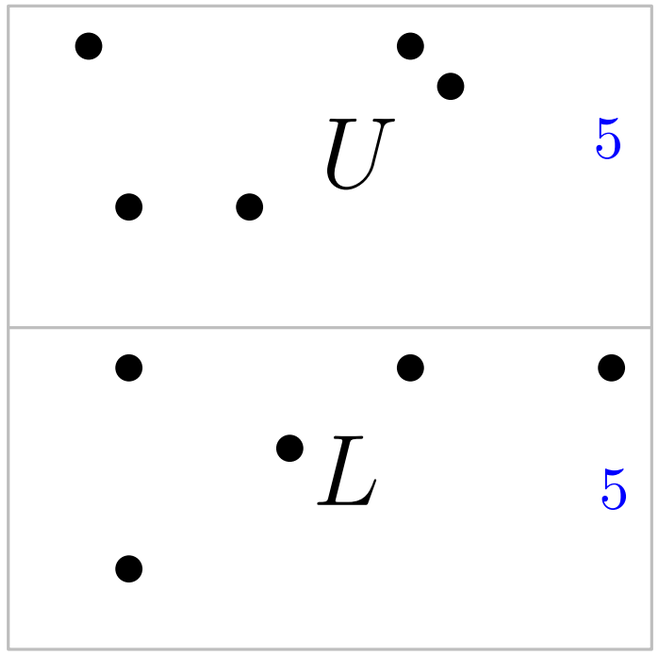
$$\frac{n_U - n_L}{n_U + n_L}$$

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Balance [Tamassia *et al.* 1988]



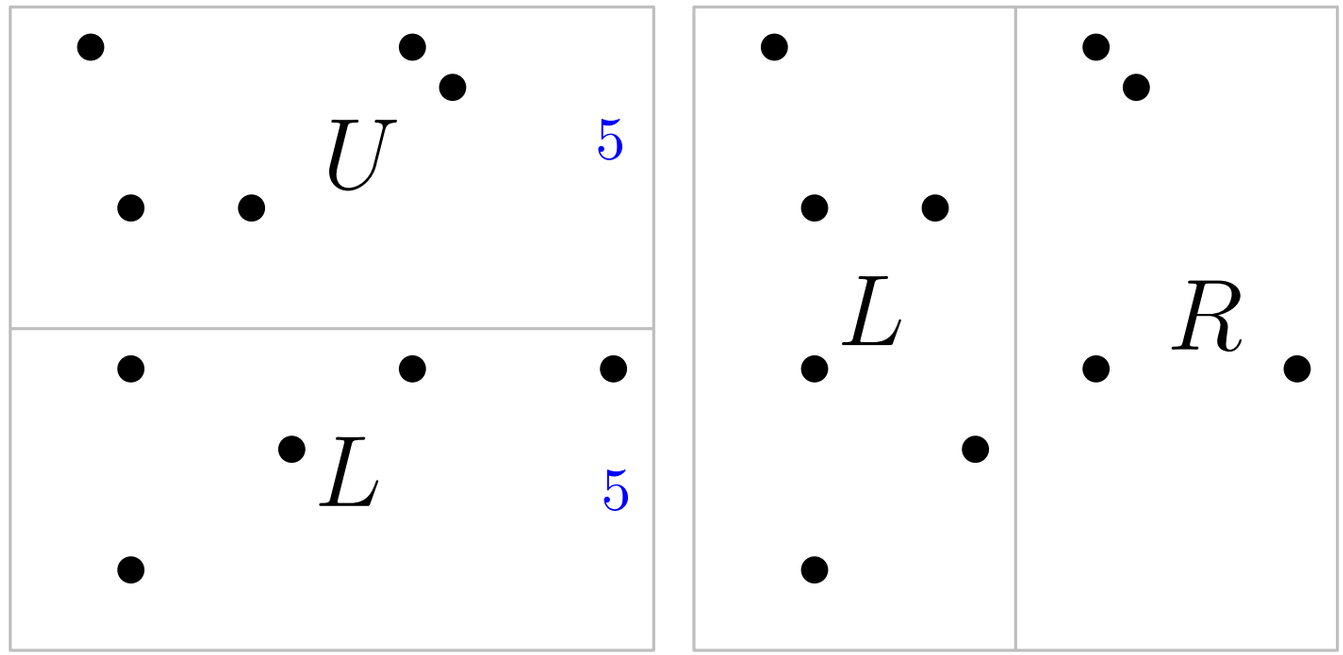
$$\frac{n_U - n_L}{n_U + n_L}$$

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Balance [Tamassia *et al.* 1988]



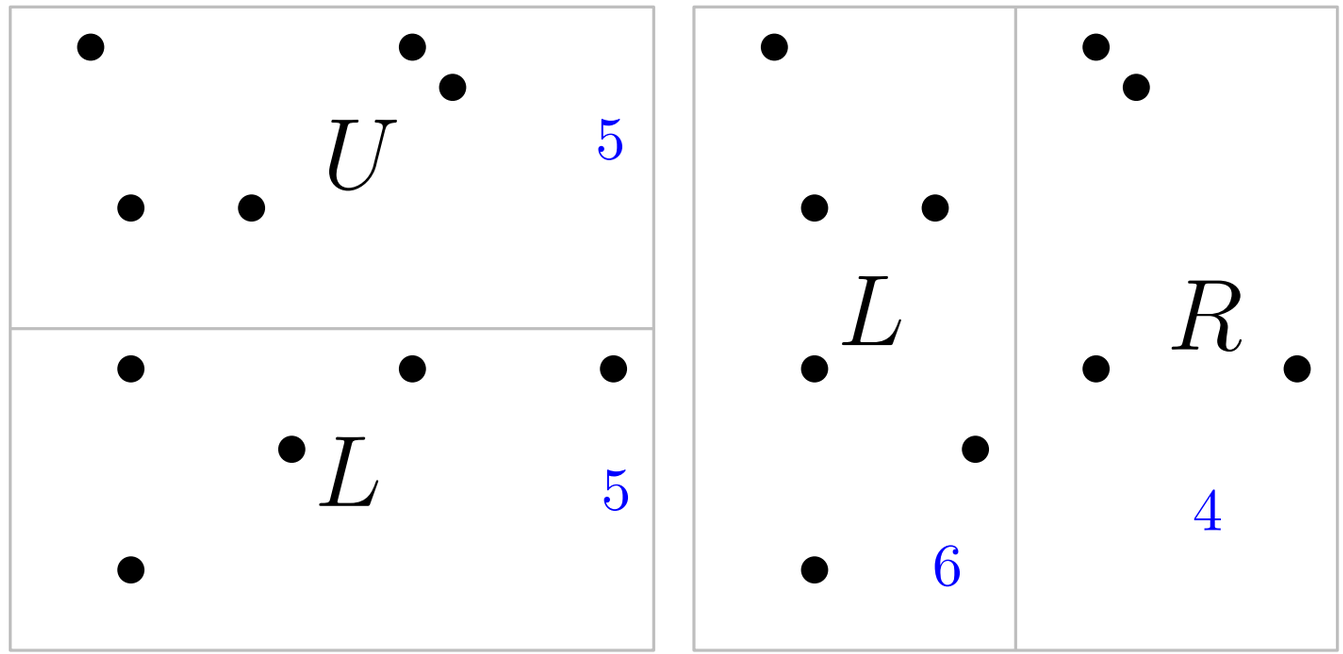
$$\frac{n_U - n_L}{n_U + n_L} \qquad \frac{n_L - n_R}{n_L + n_R}$$

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Balance [Tamassia *et al.* 1988]



$$\frac{n_U - n_L}{n_U + n_L} \qquad \frac{n_L - n_R}{n_L + n_R}$$

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Node orthogonality [Purchase 2002]

'grid-likeness'

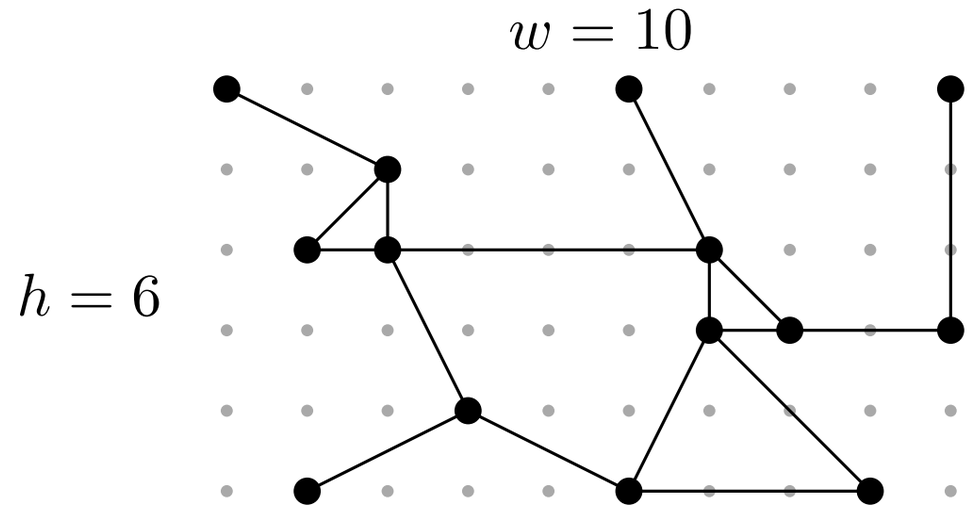
Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Node orthogonality [Purchase 2002]

'grid-likeness'



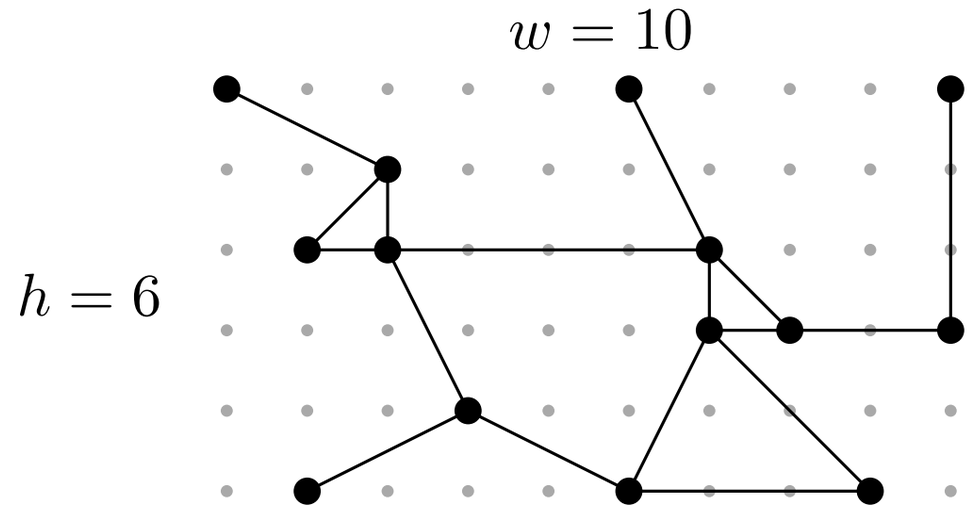
Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
~~Node orthogonality~~
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Node orthogonality [Purchase 2002]

'grid-likeness'



$$\frac{n}{w \cdot h}$$

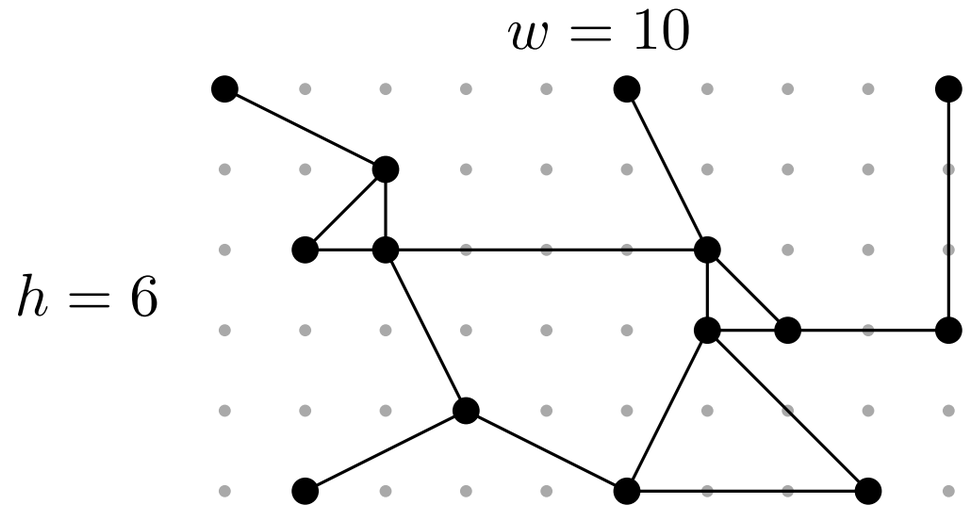
Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
~~Node orthogonality~~
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Node orthogonality [Purchase 2002]

'grid-likeness'



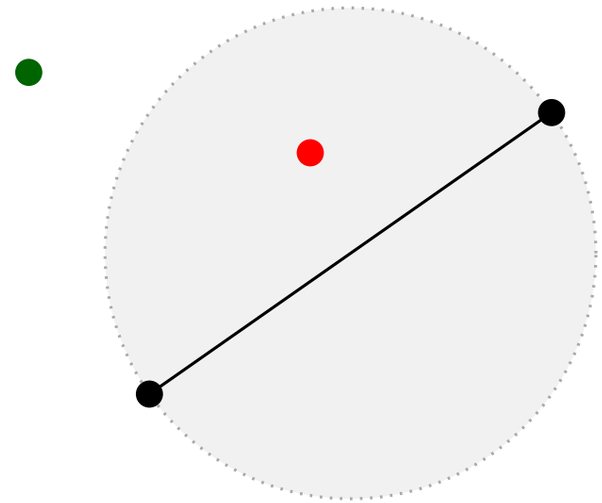
$$\frac{n}{w \cdot h} = \frac{14}{60} = 0.23$$

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Gabriel ratio [Mooney *et al.* 2024]

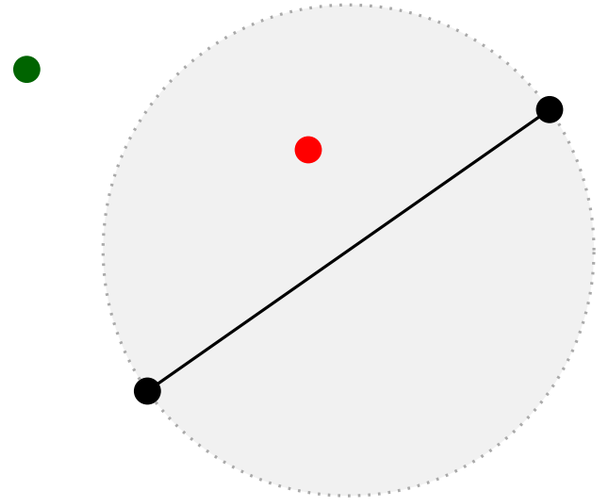


Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Gabriel ratio [Mooney *et al.* 2024]



$$1 - \frac{g}{g_{\text{max}}}$$

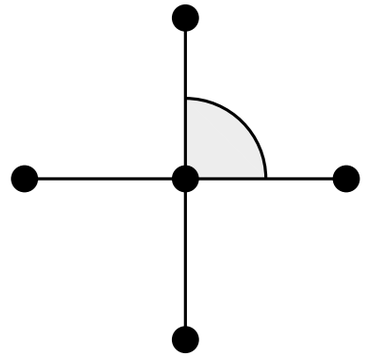
g ... number of violations
g_{max} ... $m \cdot (n-2)$

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Angular resolution [Purchase 2002]



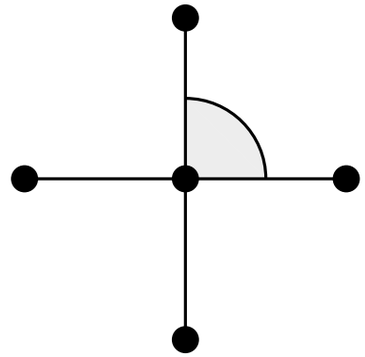
$$o_i = \frac{360°}{\deg(v_i)}$$

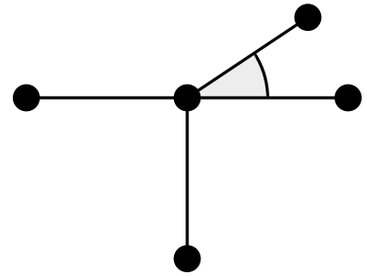
Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Angular resolution [Purchase 2002]





$$o_i = \frac{360^\circ}{\deg(v_i)}$$

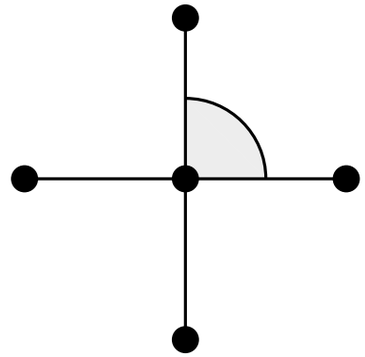
$$\theta_i^{\min}$$

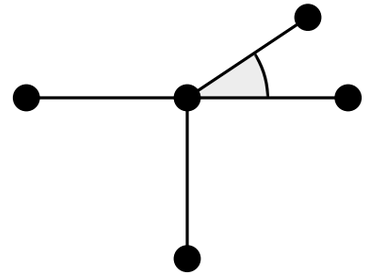
Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Angular resolution [Purchase 2002]





$$o_i = \frac{360°}{\deg(v_i)} \qquad \theta_i^{\min}$$

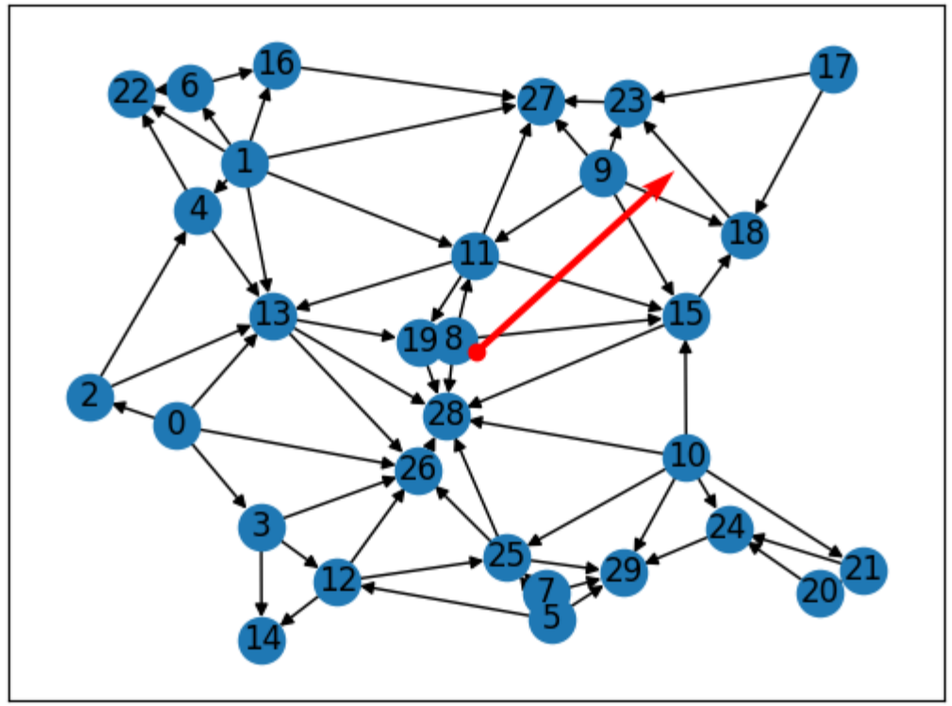
$$1 - \frac{1}{n_{\deg>1}} \sum_{i=1}^{n_{\deg>1}} \left| \frac{o_i - \theta_i^{\min}}{o_i} \right|$$

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Average flow [Purchase 2002]



$$\text{norm}\left(\sum_{e \in E} \text{norm}(\vec{e})\right)$$

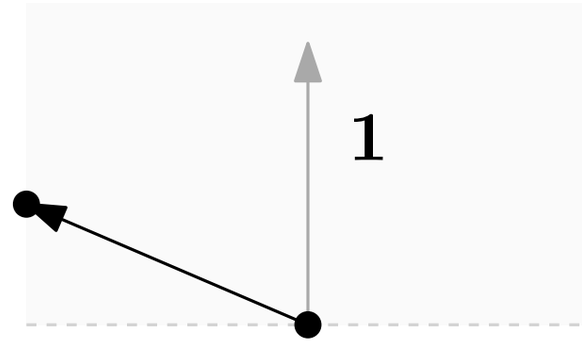
Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Upwards flow [Purchase 2002]

$$\frac{1}{m}\sum_{1}^{m}\begin{cases}1 & \text{if } \langle e_i \cdot \mathbf{1}\rangle > 0\\ 0 & \text{otherwise}\end{cases}$$



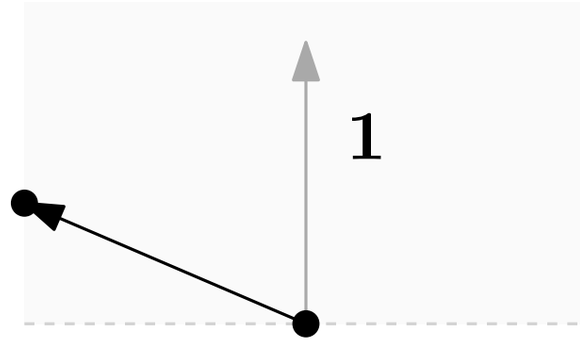
Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Upwards flow [Purchase 2002]

$$\frac{1}{m}\sum_{1}^{m}\begin{cases}1 & \text{if } \langle e_i \cdot \mathbf{1}\rangle > 0\\ 0 & \text{otherwise}\end{cases}$$



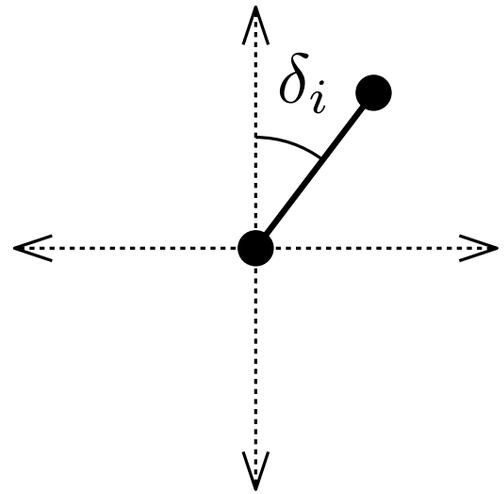
For directed graphs only

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Edge orthogonality

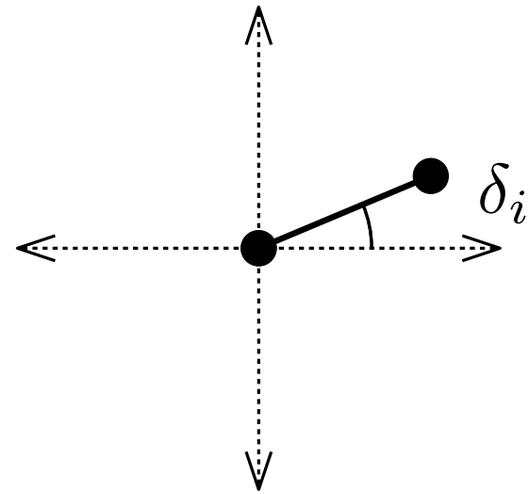


Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Edge orthogonality

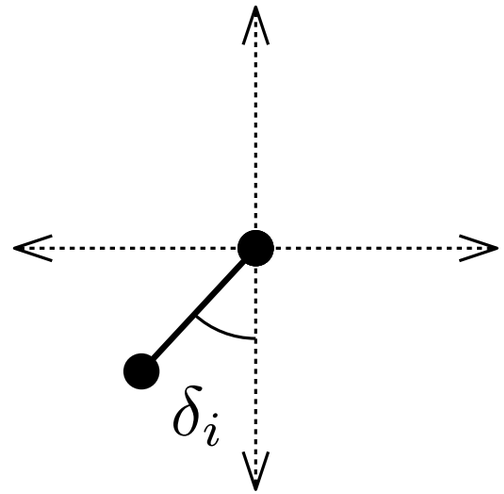


Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Edge orthogonality

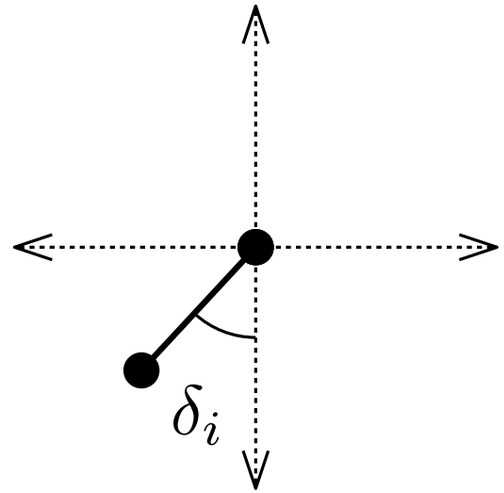


Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Edge orthogonality



$$1 - \frac{1}{m} \sum_{i=1}^{m} \frac{\delta_i}{45°}$$

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Boundary
Area
Aspect ratio

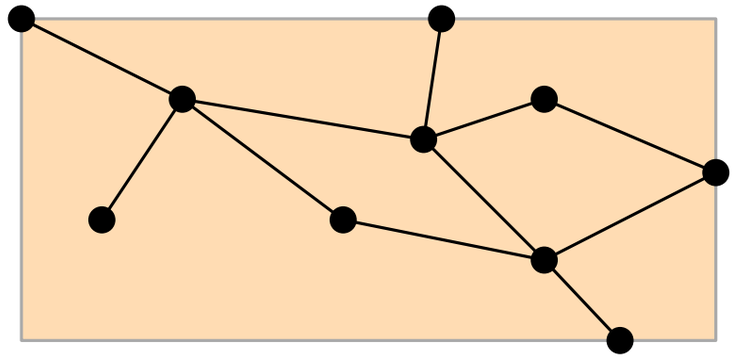
Crossings
Crossings
Crossing angles

Symmetry
Node-based
Edge-based
Visual

Metrics

Area [Taylor, Rodgers, 2005]

Bounding box



Boundary
Area
Aspect ratio

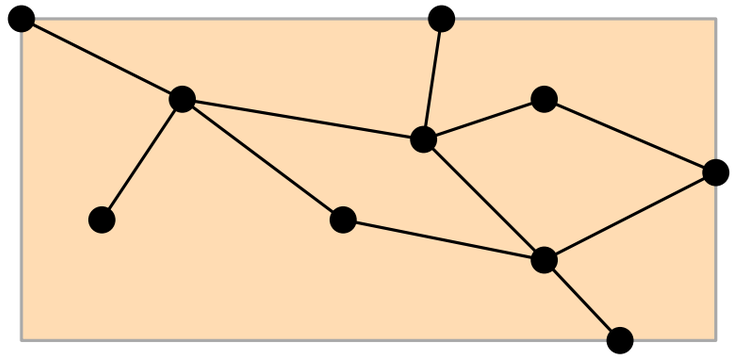
Crossings
Crossings
Crossing angles

Symmetry
Node-based
Edge-based
Visual

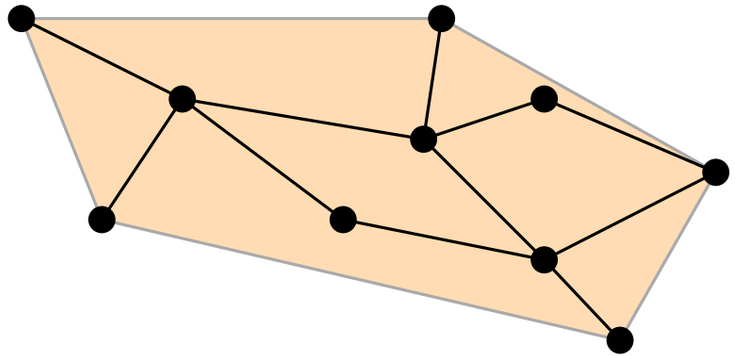
Metrics

Area [Taylor, Rodgers, 2005]

Bounding box



Convex hull



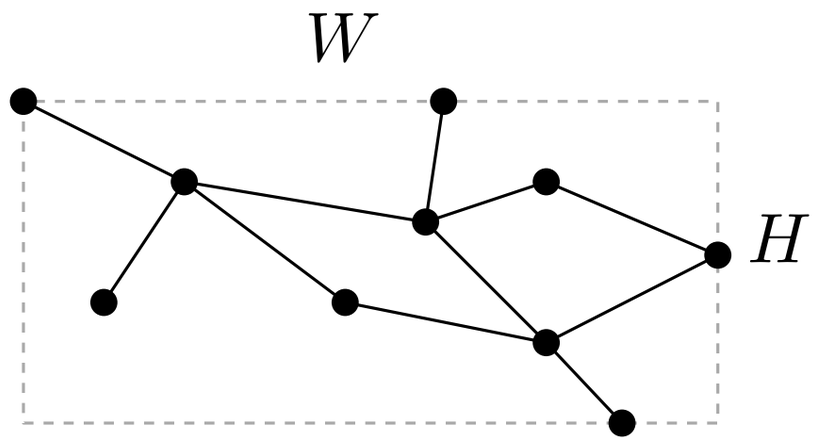
Boundary
Area
Aspect ratio

Crossings
Crossings
Crossing angles

Symmetry
Node-based
Edge-based
Visual

Metrics

Aspect ratio [Mooney *et al.* 2024]



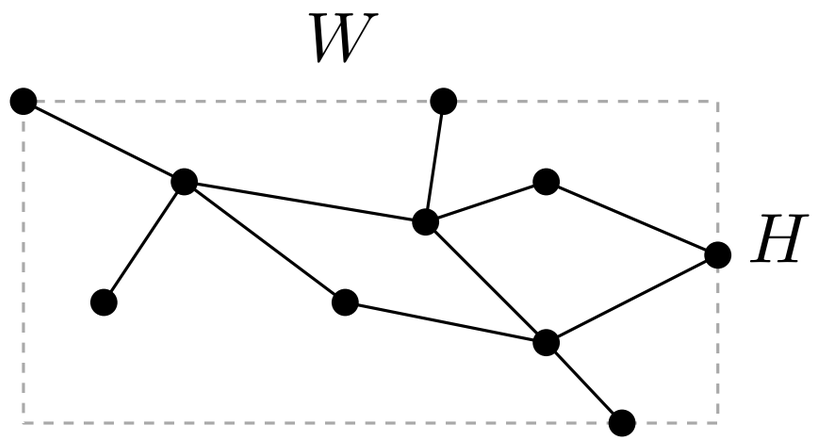
Boundary
Area
~~Aspect ratio~~

Crossings
Crossings
Crossing angles

Symmetry
Node-based
Edge-based
Visual

Metrics

Aspect ratio [Mooney *et al.* 2024]



$$\frac{\min(W,H)}{\max(W,H)}$$

Boundary
Area
~~Aspect ratio~~

Crossings
Crossings
Crossing angles

Symmetry
Node-based
Edge-based
Visual

Metrics

Crossings [Purchase 2002] [Mooney *et al.* 2024]

$O((n+k)\log n)$ Bentley Ottman Sweepline algorithm

Boundary
Area
Aspect ratio

Crossings
Crossings
Crossing angles

Symmetry
Node-based
Edge-based
Visual

Metrics

Crossings [Purchase 2002] [Mooney *et al.* 2024]

$O((n+k)\log n)$ Bentley Ottman Sweepline algorithm

$$1 - \frac{c}{c_{\text{max}}}$$

Boundary
Area
Aspect ratio

Crossings
Crossings
Crossing angles

Symmetry
Node-based
Edge-based
Visual

Metrics

Crossings [Purchase 2002] [Mooney *et al.* 2024]

$O((n+k)\log n)$ Bentley Ottman Sweepline algorithm

$$1 - \frac{c}{c_{\text{max}}}$$

$$c_{\text{max}} = \frac{m(m-1)}{2} - c_{\text{deg}} - c_{\text{tri}} - c_{\text{4cyc}}$$

Boundary
Area
Aspect ratio

Crossings
Crossings
Crossing angles

Symmetry
Node-based
Edge-based
Visual

Metrics

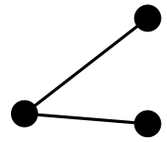
Crossings [Purchase 2002] [Mooney *et al.* 2024]

$O((n+k)\log n)$ Bentley Ottman Sweepline algorithm

$$1 - \frac{c}{c_{\text{max}}}$$

$$c_{\text{max}} = \frac{m(m-1)}{2} - c_{\text{deg}} - c_{\text{tri}} - c_{\text{4cyc}}$$

c_{deg} . . . neighbouring edges



Boundary
Area
Aspect ratio

Crossings
Crossings
Crossing angles

Symmetry
Node-based
Edge-based
Visual

Metrics

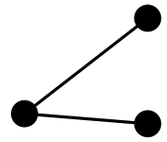
Crossings [Purchase 2002] [Mooney *et al.* 2024]

$O((n+k)\log n)$ Bentley Ottman Sweepline algorithm

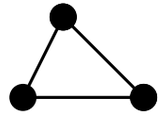
$$1 - \frac{c}{c_{\text{max}}}$$

$$c_{\text{max}} = \frac{m(m-1)}{2} - c_{\text{deg}} - c_{\text{tri}} - c_{\text{4cyc}}$$

c_{deg} ... neighbouring edges



c_{tri} ... triangles



Boundary
Area
Aspect ratio

Crossings
Crossings
Crossing angles

Symmetry
Node-based
Edge-based
Visual

Metrics

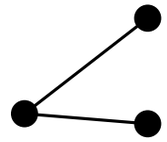
Crossings [Purchase 2002] [Mooney *et al.* 2024]

$O((n+k)\log n)$ Bentley Ottman Sweepline algorithm

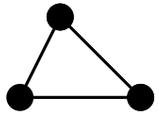
$$1 - \frac{c}{c_{\text{max}}}$$

$$c_{\text{max}} = \frac{m(m-1)}{2} - c_{\text{deg}} - c_{\text{tri}} - c_{\text{4cyc}}$$

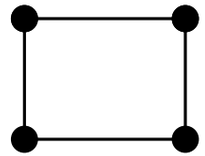
c_{deg} ... neighbouring edges

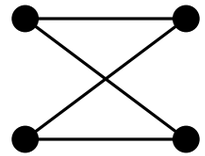


c_{tri} ... triangles



c_{4cyc} ... cycles of length 4





Boundary
Area
Aspect ratio

Crossings
Crossings
Crossing angles

Symmetry
Node-based
Edge-based
Visual

Metrics

Crossing angles

Angular resolution for crossings

Boundary
Area
Aspect ratio

Crossings
Crossings
Crossing angles

Symmetry
Node-based
Edge-based
Visual

Metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Boundary
Area
Aspect ratio

Crossings
Crossings
Crossing angles

Symmetry
Node-based
Edge-based
Visual

Symmetry

	Type	**Runtime**
Node-based	Reflective	$O(n^7)$
Edge-based	Reflective, rotational, translational	$O(m^2 + N)$
Stress-based	n.a.	$O(n \log n)$
Force-based	n.a.	$O(n \log n)$
Machine Learning	Reflective	$O(n + m)$
Automorphism detection	Reflective, translational	$O(n \log n)$

Symmetry

	Type	**Runtime**
Node-based	Reflective	$O(n^7)$
Edge-based	Reflective, rotational, translational	$O(m^2 + N)$
Stress-based	n.a.	$O(n \log n)$
Force-based	n.a.	$O(n \log n)$
Machine Learning	Reflective	$O(n + m)$
Automorphism detection	Reflective, translational	$O(n \log n)$

Symmetry

	Type	**Runtime**
Node-based	Reflective	$O(n^7)$
Edge-based	Reflective, rotational, translational	$O(m^2 + N)$
Stress-based	n.a.	$O(n \log n)$
Force-based	n.a.	$O(n \log n)$
Machine Learning	Reflective	$O(n + m)$
Automorphism detection	Reflective, translational	$O(n \log n)$
Visual symmetry	Reflective, rotational, translational	$O(n + m)$

Symmetry

	Type	Runtime
Node-based	Reflective	$O(n^7)$
Edge-based	Reflective, rotational, translational	$O(m^2 + N)$
Stress-based	n.a.	$O(n \log n)$
Force-based	n.a.	$O(n \log n)$
Machine Learning	Reflective	$O(n + m)$
Automorphism detection	Reflective, translational	$O(n \log n)$
Visual symmetry	Reflective, rotational, translational	$O(n + m)$

Node-based symmetry [Purchase 2002]

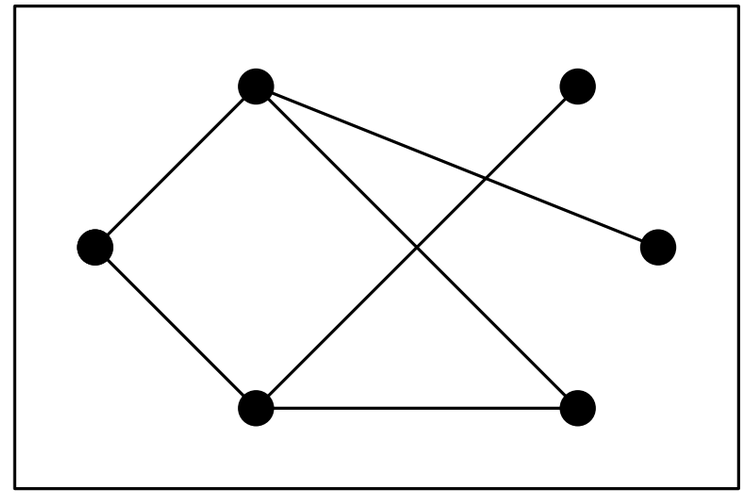
- Reflective symmetry only

Node-based symmetry [Purchase 2002]

- Reflective symmetry only
- Primarily concerned about node positions

Node-based symmetry [Purchase 2002]

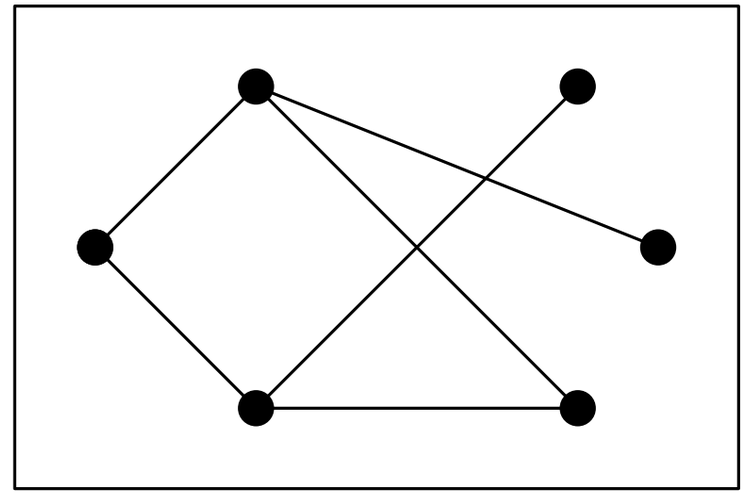
- Reflective symmetry only
- Primarily concerned about node positions



Node-based symmetry [Purchase 2002]

- Reflective symmetry only
- Primarily concerned about node positions

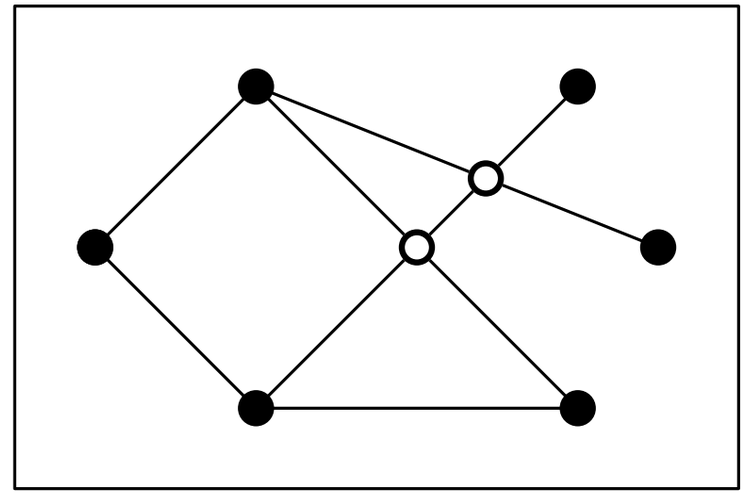
1. Elevate crossings to nodes



Node-based symmetry [Purchase 2002]

- Reflective symmetry only
- Primarily concerned about node positions

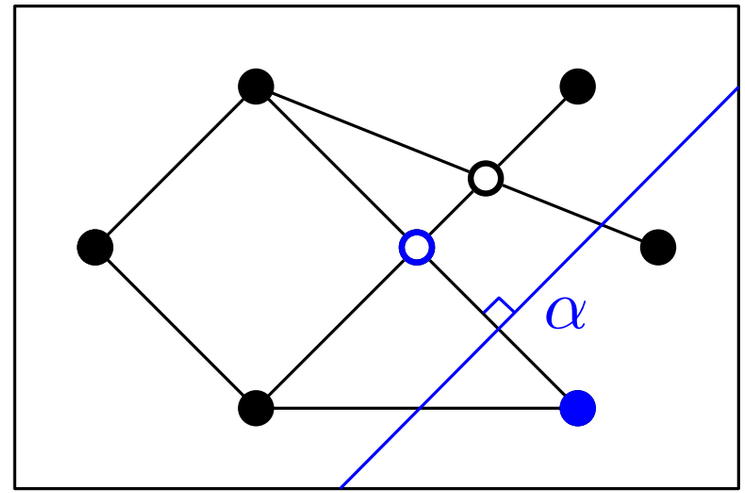
1. Elevate crossings to nodes



Node-based symmetry [Purchase 2002]

- Reflective symmetry only
- Primarily concerned about node positions

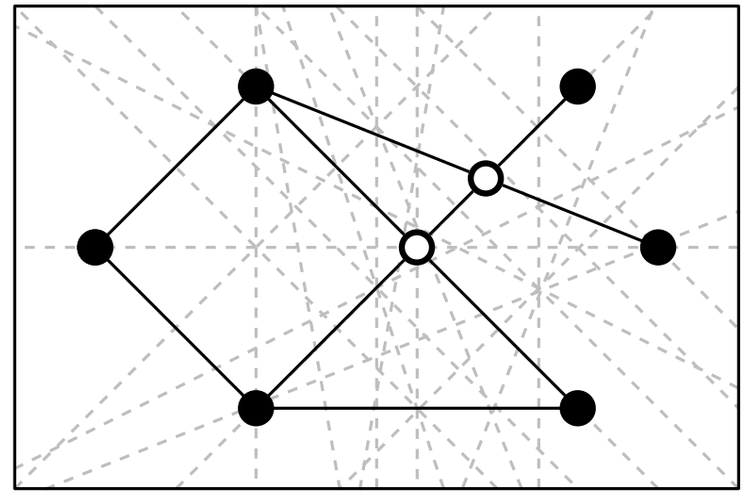
1. Elevate crossings to nodes
2. Create an axis α at perpenticular bisector of each pair of nodes



Node-based symmetry [Purchase 2002]

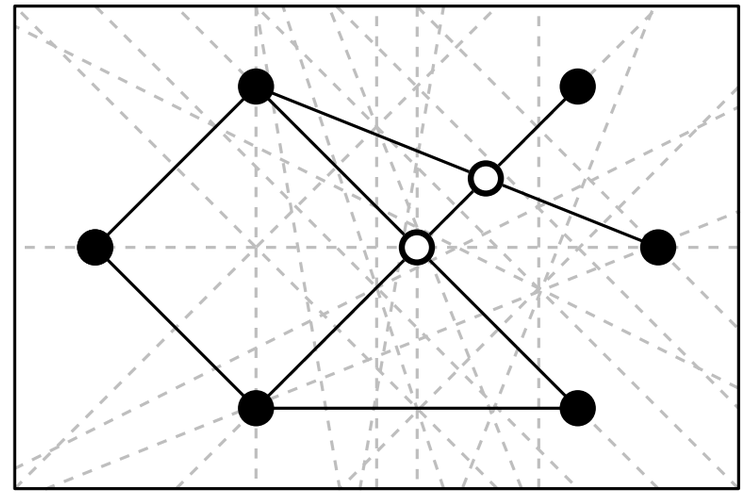
- Reflective symmetry only
- Primarily concerned about node positions

1. Elevate crossings to nodes
2. Create an axis α at perpenticular bisector of each pair of nodes



Node-based symmetry [Purchase 2002]

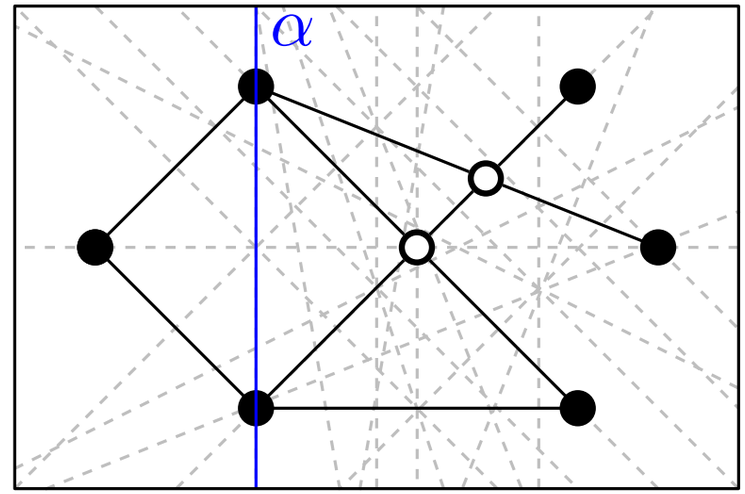
- Reflective symmetry only
- Primarily concerned about node positions



1. Elevate crossings to nodes
2. Create an axis α at perpenticular bisector of each pair of nodes
3. For each axis α:

Node-based symmetry [Purchase 2002]

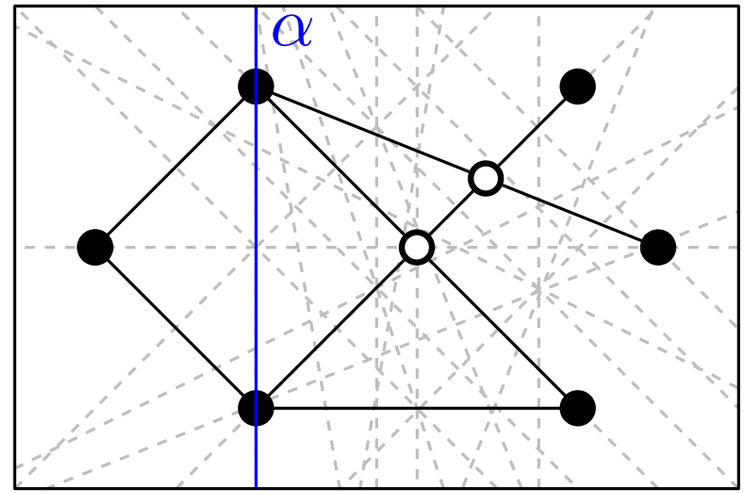
- Reflective symmetry only
- Primarily concerned about node positions



1. Elevate crossings to nodes
2. Create an axis α at perpenticular bisector of each pair of nodes
3. For each axis α:

Node-based symmetry [Purchase 2002]

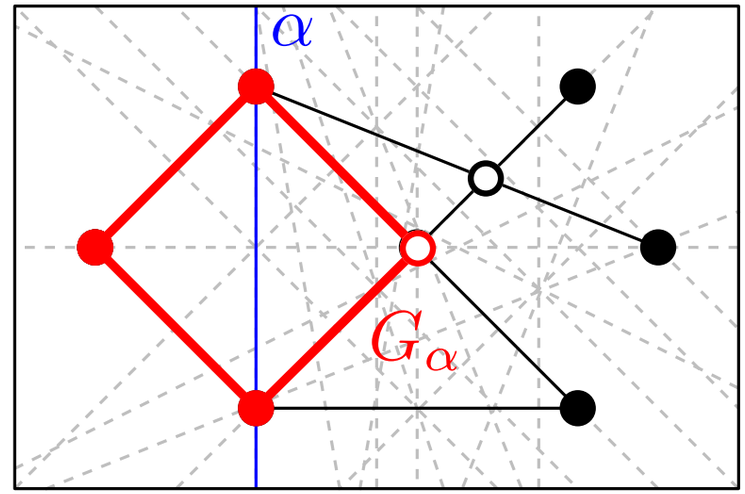
- Reflective symmetry only
- Primarily concerned about node positions



1. Elevate crossings to nodes
2. Create an axis α at perpenticular bisector of each pair of nodes
3. For each axis α:
   - Obtain subgraph G_α containing all mirrored nodes

Node-based symmetry [Purchase 2002]

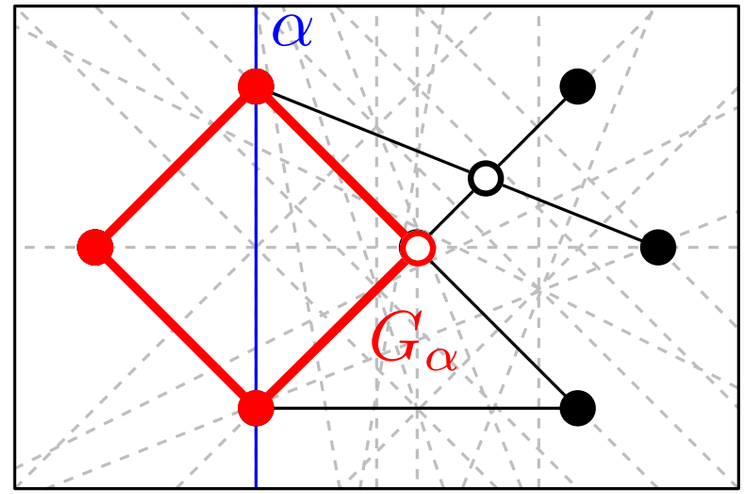
- Reflective symmetry only
- Primarily concerned about node positions



1. Elevate crossings to nodes
2. Create an axis α at perpenticular bisector of each pair of nodes
3. For each axis α:
   - Obtain subgraph G_α containing all mirrored nodes

Node-based symmetry [Purchase 2002]

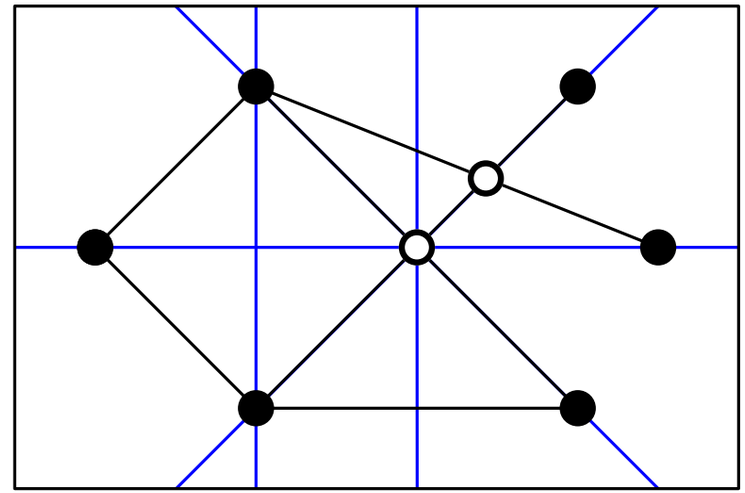
- Reflective symmetry only
- Primarily concerned about node positions



1. Elevate crossings to nodes
2. Create an axis α at perpenticular bisector of each pair of nodes
3. For each axis α:
   - Obtain subgraph G_α containing all mirrored nodes
   - If $|E(G_\alpha)| >$ THRESHOLD: keep α

Node-based symmetry [Purchase 2002]

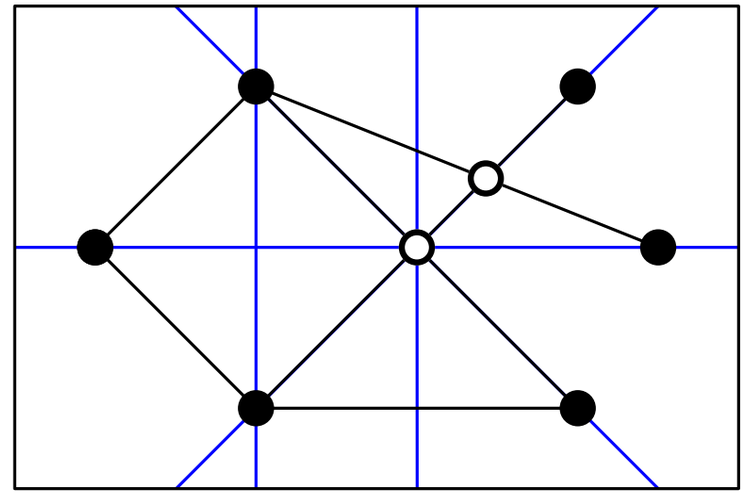
- Reflective symmetry only
- Primarily concerned about node positions



1. Elevate crossings to nodes
2. Create an axis α at perpenticular bisector of each pair of nodes
3. For each axis α:
   - Obtain subgraph G_α containing all mirrored nodes
   - If $|E(G_\alpha)| >$ THRESHOLD: keep α

Node-based symmetry [Purchase 2002]

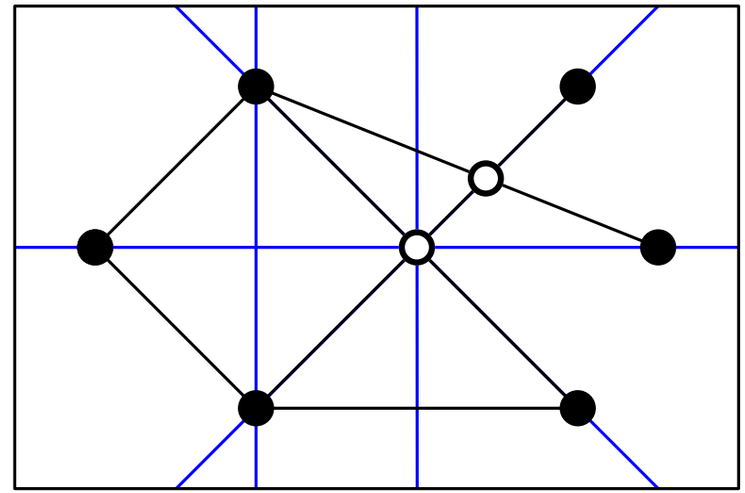
- Reflective symmetry only
- Primarily concerned about node positions



1. Elevate crossings to nodes
2. Create an axis α at perpenticular bisector of each pair of nodes
3. For each axis α:
   - Obtain subgraph G_α containing all mirrored nodes
   - If $|E(G_\alpha)| >$ THRESHOLD: keep α
4. $\forall\alpha$: Obtain symmetry estimate s_α

Node-based symmetry [Purchase 2002]

- Reflective symmetry only
- Primarily concerned about node positions

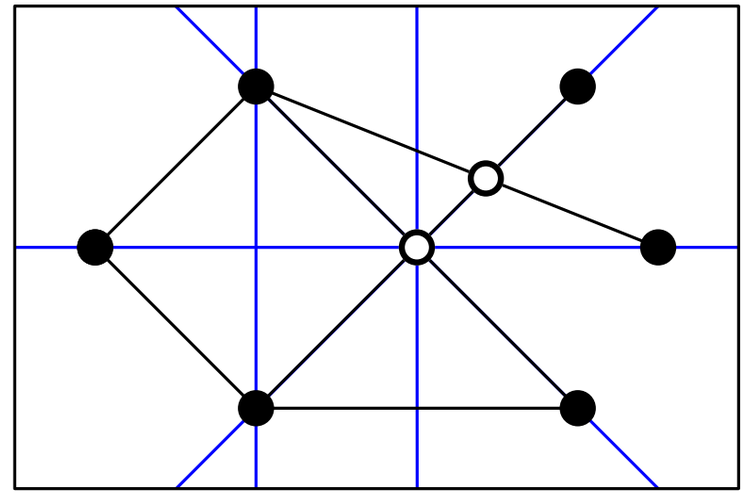


1. Elevate crossings to nodes
2. Create an axis α at perpenticular bisector of each pair of nodes
3. For each axis α:
   - Obtain subgraph G_α containing all mirrored nodes
   - If $|E(G_\alpha)| >$ THRESHOLD: keep α
4. $\forall\alpha$: Obtain symmetry estimate s_α

Next slide

Node-based symmetry [Purchase 2002]

- Reflective symmetry only
- Primarily concerned about node positions



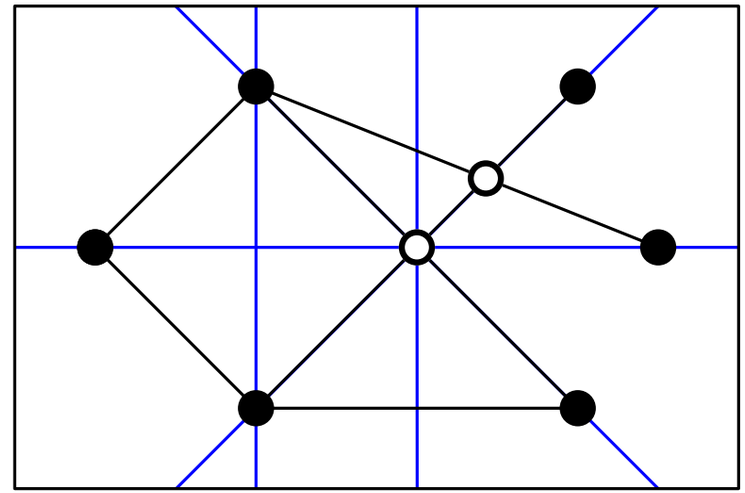
1. Elevate crossings to nodes
2. Create an axis α at perpenticular bisector of each pair of nodes
3. For each axis α:
   - Obtain subgraph G_α containing all mirrored nodes
   - If $|E(G_\alpha)| >$ THRESHOLD: keep α
4. $\forall\alpha$: Obtain symmetry estimate s_α
5. Final metric:

Next slide

$$\frac{\sum_\alpha s_\alpha \cdot \text{area}(G_\alpha)}{\max(\text{area}(G), \sum_\alpha \text{area}(G_\alpha))}$$

Node-based symmetry [Purchase 2002]

- Reflective symmetry only
- Primarily concerned about node positions

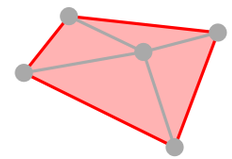


1. Elevate crossings to nodes
2. Create an axis α at perpenticular bisector of each pair of nodes
3. For each axis α:
   - Obtain subgraph G_α containing all mirrored nodes
   - If $|E(G_\alpha)| >$ THRESHOLD: keep α
4. $\forall\alpha$: Obtain symmetry estimate s_α
5. Final metric:

Next slide

$$\frac{\sum_\alpha s_\alpha \cdot \text{area}(G_\alpha)}{\max(\text{area}(G), \sum_\alpha \text{area}(G_\alpha))}$$

Convex Hull



Node-based symmetry [Purchase 2002]

- Reflective symmetry only
- Primarily concerned about node positions

1. Elevate crossings to nodes
2. Create an axis α at perpenticular bisector of each pair of nodes
3. For each axis α:
   - Obtain subgraph G_α containing all mirrored nodes
   - If $|E(G_\alpha)| >$ THRESHOLD: keep α
4. $\forall\alpha$: Obtain symmetry estimate s_α
5. Final metric:

$$\frac{\sum_\alpha s_\alpha \cdot \text{area}(G_\alpha)}{\max(\text{area}(G), \sum_\alpha \text{area}(G_\alpha))}$$

Next slide

Convex Hull

Node-based symmetry [Purchase 2002]

4. $\forall \alpha$: Obtain symmetry estimate s_α

Node-based symmetry [Purchase 2002]

4. $\forall \alpha$: Obtain symmetry estimate s_α

   $\forall e, e' \in E(G_\alpha)$ reflected over α:

Node-based symmetry [Purchase 2002]

4. $\forall \alpha$: Obtain symmetry estimate s_α

   $\forall e, e' \in E(G_\alpha)$ reflected over α:

$$s_\alpha^{e,e'} = \begin{cases} f^2 & \text{if both pairs of endpoints are of different types} \\ f & \text{if one pair consists of different types} \\ 1 & \text{otherwise,} \end{cases}$$

   $f \in [0, 1] \ldots$ parameter for similarity between crossings and nodes

Node-based symmetry [Purchase 2002]

4. $\forall \alpha$: Obtain symmetry estimate s_α

   $\forall e, e' \in E(G_\alpha)$ reflected over α:

$$s_\alpha^{e,e'} = \begin{cases} f^2 & \text{if both pairs of endpoints are of different types} \\ f & \text{if one pair consists of different types} \\ 1 & \text{otherwise,} \end{cases}$$

   $f \in [0, 1] \ldots$ parameter for similarity between crossings and nodes

   $s_\alpha \ldots$ average $s_\alpha^{e,e'}$ over all reflected pairs of edges

Node-based symmetry [Purchase 2002]

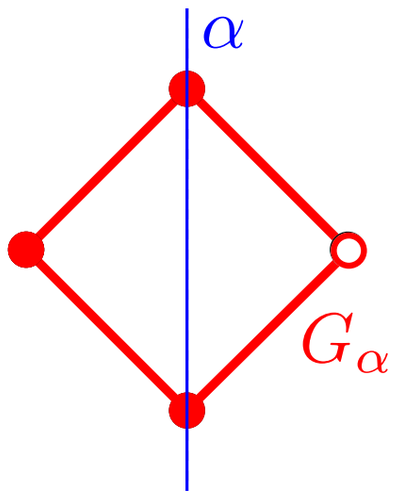
4. $\forall \alpha$: Obtain symmetry estimate s_α

   $\forall e, e' \in E(G_\alpha)$ reflected over α:

$$s_\alpha^{e,e'} = \begin{cases} f^2 & \text{if both pairs of endpoints are of different types} \\ f & \text{if one pair consists of different types} \\ 1 & \text{otherwise,} \end{cases}$$

   $f \in [0,1] \ldots$ parameter for similarity between crossings and nodes

   $s_\alpha \ldots$ average $s_\alpha^{e,e'}$ over all reflected pairs of edges



Node-based symmetry [Purchase 2002]

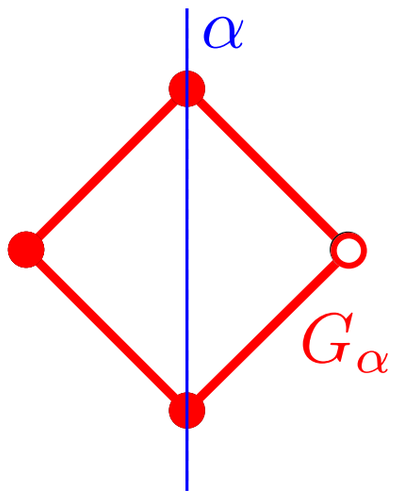
4. $\forall \alpha$: Obtain symmetry estimate s_α

$\forall e, e' \in E(G_\alpha)$ reflected over α:

$$s_\alpha^{e,e'} = \begin{cases} f^2 & \text{if both pairs of endpoints are of different types} \\ f & \text{if one pair consists of different types} \\ 1 & \text{otherwise,} \end{cases}$$

$f \in [0,1] \ldots$ parameter for similarity between crossings and nodes

$s_\alpha \ldots$ average $s_\alpha^{e,e'}$ over all reflected pairs of edges



$f := 0.5$

Node-based symmetry [Purchase 2002]

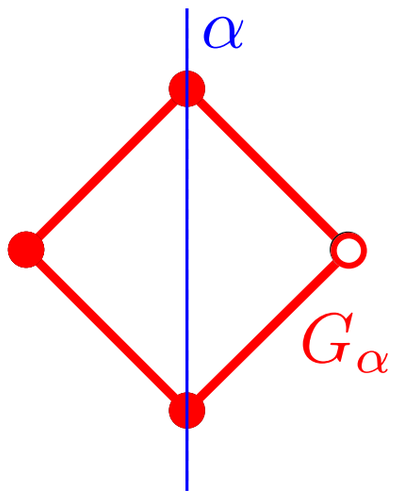
4. $\forall \alpha$: Obtain symmetry estimate s_α

   $\forall e, e' \in E(G_\alpha)$ reflected over α:

$$s_\alpha^{e,e'} = \begin{cases} f^2 & \text{if both pairs of endpoints are of different types} \\ f & \text{if one pair consists of different types} \\ 1 & \text{otherwise,} \end{cases}$$

   $f \in [0,1] \ldots$ parameter for similarity between crossings and nodes

   $s_\alpha \ldots$ average $s_\alpha^{e,e'}$ over all reflected pairs of edges



$f := 0.5$

$s_\alpha = \text{avg}($ $)$

Node-based symmetry [Purchase 2002]

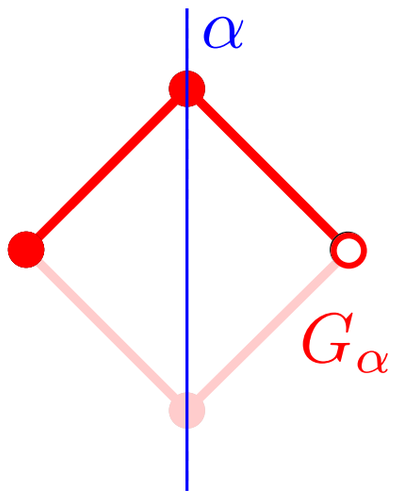
4. $\forall \alpha$: Obtain symmetry estimate s_α

$\forall e, e' \in E(G_\alpha)$ reflected over α:

$$s_\alpha^{e,e'} = \begin{cases} f^2 & \text{if both pairs of endpoints are of different types} \\ f & \text{if one pair consists of different types} \\ 1 & \text{otherwise,} \end{cases}$$

$f \in [0, 1] \ldots$ parameter for similarity between crossings and nodes

$s_\alpha \ldots$ average $s_\alpha^{e,e'}$ over all reflected pairs of edges



$f := 0.5$

$s_\alpha = \text{avg}($ $)$

Node-based symmetry [Purchase 2002]

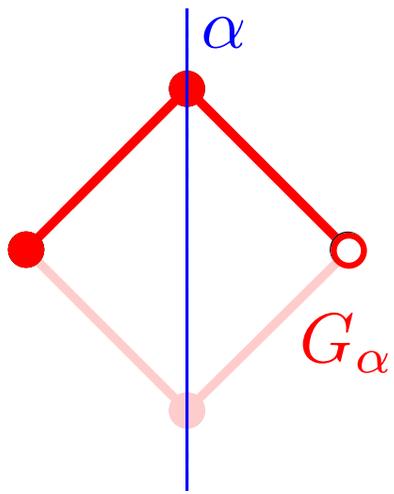
4. $\forall \alpha$: Obtain symmetry estimate s_α

$\forall e, e' \in E(G_\alpha)$ reflected over α:

$$s_\alpha^{e,e'} = \begin{cases} f^2 & \text{if both pairs of endpoints are of different types} \\ f & \text{if one pair consists of different types} \\ 1 & \text{otherwise,} \end{cases}$$

$f \in [0, 1] \ldots$ parameter for similarity between crossings and nodes

$s_\alpha \ldots$ average $s_\alpha^{e,e'}$ over all reflected pairs of edges



$f := 0.5$

$s_\alpha = \text{avg}(0.5, \quad)$

Node-based symmetry [Purchase 2002]

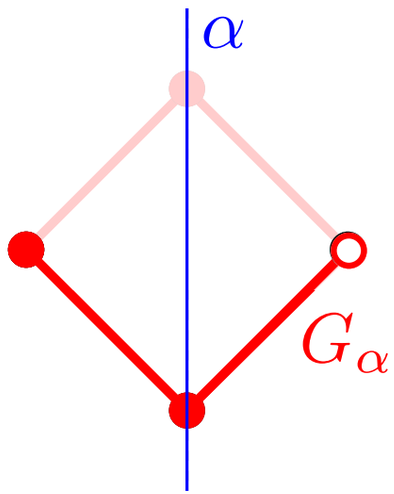
4. $\forall \alpha$: Obtain symmetry estimate s_α

$\forall e, e' \in E(G_\alpha)$ reflected over α:

$$s_\alpha^{e,e'} = \begin{cases} f^2 & \text{if both pairs of endpoints are of different types} \\ f & \text{if one pair consists of different types} \\ 1 & \text{otherwise,} \end{cases}$$

$f \in [0, 1] \ldots$ parameter for similarity between crossings and nodes

$s_\alpha \ldots$ average $s_\alpha^{e,e'}$ over all reflected pairs of edges



$f := 0.5$

$s_\alpha = \text{avg}(0.5, 0.5)$

Node-based symmetry [Purchase 2002]

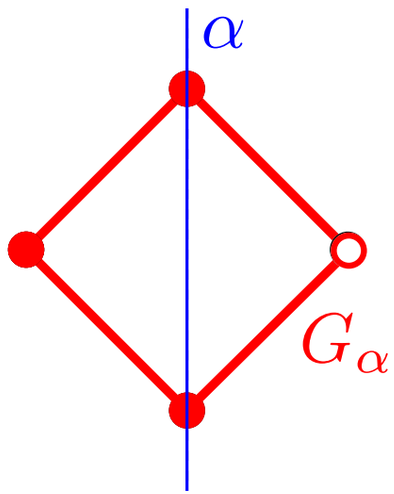
4. $\forall \alpha$: Obtain symmetry estimate s_α

$\forall e, e' \in E(G_\alpha)$ reflected over α:

$$s_\alpha^{e,e'} = \begin{cases} f^2 & \text{if both pairs of endpoints are of different types} \\ f & \text{if one pair consists of different types} \\ 1 & \text{otherwise,} \end{cases}$$

$f \in [0,1] \ldots$ parameter for similarity between crossings and nodes

$s_\alpha \ldots$ average $s_\alpha^{e,e'}$ over all reflected pairs of edges

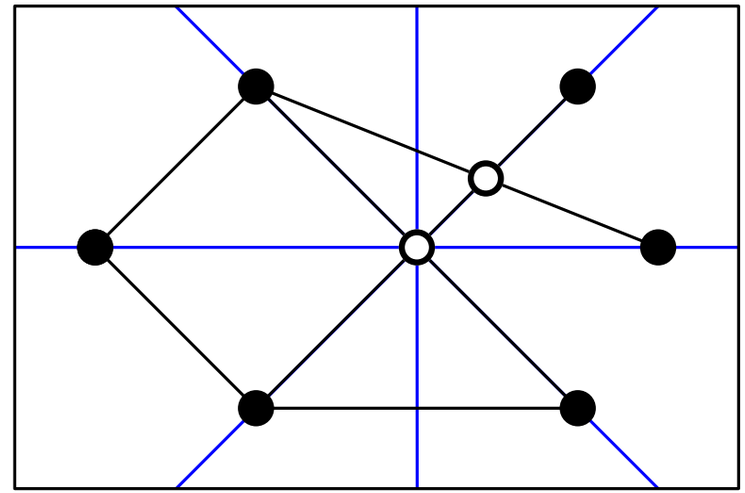


$f := 0.5$

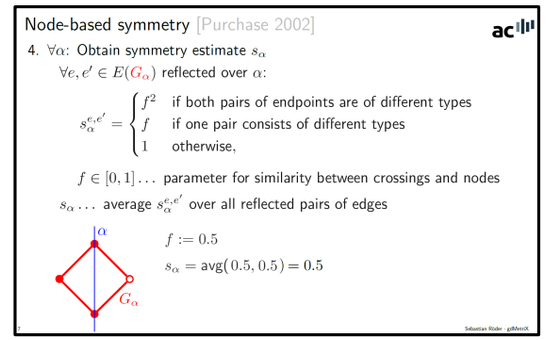
$s_\alpha = \text{avg}(0.5, 0.5) = 0.5$

Node-based symmetry [Purchase 2002]

- Reflective symmetry only
- Primarily concerned about node positions

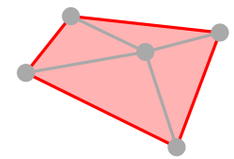


1. Elevate crossings to nodes
2. Create an axis α at perpenticular bisector of each pair of nodes
3. For each axis α:
   - Obtain subgraph G_α containing all mirrored nodes
   - If $|E(G_\alpha)| >$ THRESHOLD: keep α
4. $\forall\alpha$: Obtain symmetry estimate s_α
5. Final metric:



$$\frac{\sum_\alpha s_\alpha \cdot \text{area}(G_\alpha)}{\max(\text{area}(G), \sum_\alpha \text{area}(G_\alpha))}$$

Convex Hull



Symmetry

	Type	**Runtime**
Node-based	Reflective	$O(n^7)$
Edge-based	Reflective, rotational, translational	$O(m^2 + N)$
Stress-based	n.a.	$O(n \log n)$
Force-based	n.a.	$O(n \log n)$
Machine Learning	Reflective	$O(n + m)$
Automorphism detection	Reflective, translational	$O(n \log n)$
Visual symmetry	Reflective, rotational, translational	$O(n + m)$

Symmetry

	Type	**Runtime**
Node-based	Reflective	$O(n^7)$
Edge-based	Reflective, rotational, translational	$O(m^2 + N)$
Stress-based	n.a.	$O(n \log n)$
Force-based	n.a.	$O(n \log n)$
Machine Learning	Reflective	$O(n + m)$
Automorphism detection	Reflective, translational	$O(n \log n)$
Visual symmetry	Reflective, rotational, translational	$O(n + m)$

Edge-based symmetry [Klapauk *et al.* 2018]

- Previous approach:
  - Primarily based on nodes only
  - $O(n^7)$ runtime
  - Only reflective symmetry

Edge-based symmetry [Klapauk *et al.* 2018]

- Previous approach:
  - Primarily based on nodes only
  - $O(n^7)$ runtime
  - Only reflective symmetry

$\Rightarrow$ faster edge-based approach

Edge-based symmetry [Klapauk *et al.* 2018]

- Previous approach:
  - Primarily based on nodes only
  - $O(n^7)$ runtime
  - Only reflective symmetry

$\Rightarrow$ faster edge-based approach

- Reflective, rotational, translational symmetry

Edge-based symmetry [Klapauk *et al.* 2018]

Reflective symmetry:

1. Generate axes

Edge-based symmetry [Klapauk *et al.* 2018]

Reflective symmetry:

1. Generate axes

   $\forall e \in E(G)$:

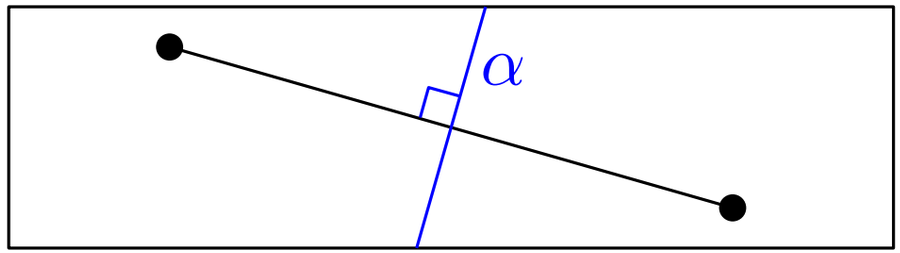
   $\forall e, e' \in E(G)$:

Edge-based symmetry [Klapauk *et al.* 2018]

Reflective symmetry:

1. Generate axes

$\forall e \in E(G)$: axis α along perpindicular bisector



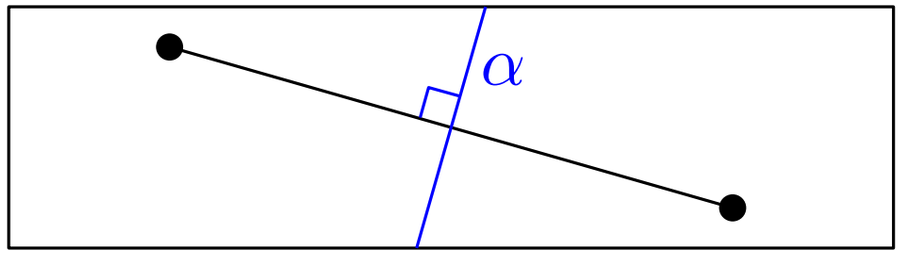
$\forall e, e' \in E(G)$:

Edge-based symmetry [Klapauk *et al.* 2018]

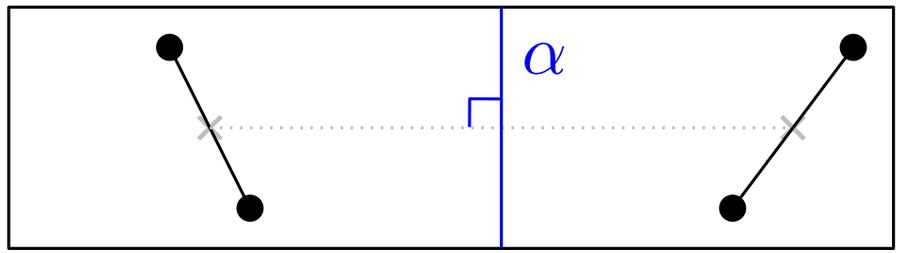
Reflective symmetry:

1. Generate axes

$\forall e \in E(G)$: axis α along perpindicular bisector



$\forall e, e' \in E(G)$: axis α along perpindicular bisector of midpoints



Edge-based symmetry [Klapauk *et al.* 2018]

Reflective symmetry:

1. Generate axes

   $\forall e \in E(G)$: perpindicular bisector

   $\forall e, e' \in E(G)$: perpindicular bisector of midpoints

Edge-based symmetry [Klapauk *et al.* 2018]

Reflective symmetry:

1. Generate axes

   $\forall e \in E(G)$: perpindicular bisector

   $\forall e, e' \in E(G)$: perpindicular bisector of midpoints

Axis score:

$s_\alpha = 1$

Edge-based symmetry [Klapauk *et al.* 2018]

Reflective symmetry:

1. Generate axes

   $\forall e \in E(G)$: perpindicular bisector

   $\forall e, e' \in E(G)$: perpindicular bisector of midpoints

Axis score:

$s_\alpha = 1$

$s_\alpha =$ edge similarity

Edge-based symmetry [Klapauk *et al.* 2018]

Reflective symmetry:

1. Generate axes

   $\forall e \in E(G)$: perpindicular bisector

   $\forall e, e' \in E(G)$: perpindicular bisector of midpoints

2. Group close axes

Axis score:

$s_\alpha = 1$

$s_\alpha =$ edge similarity

Edge-based symmetry [Klapauk *et al.* 2018]

Reflective symmetry:

1. Generate axes

   $\forall e \in E(G)$: perpindicular bisector

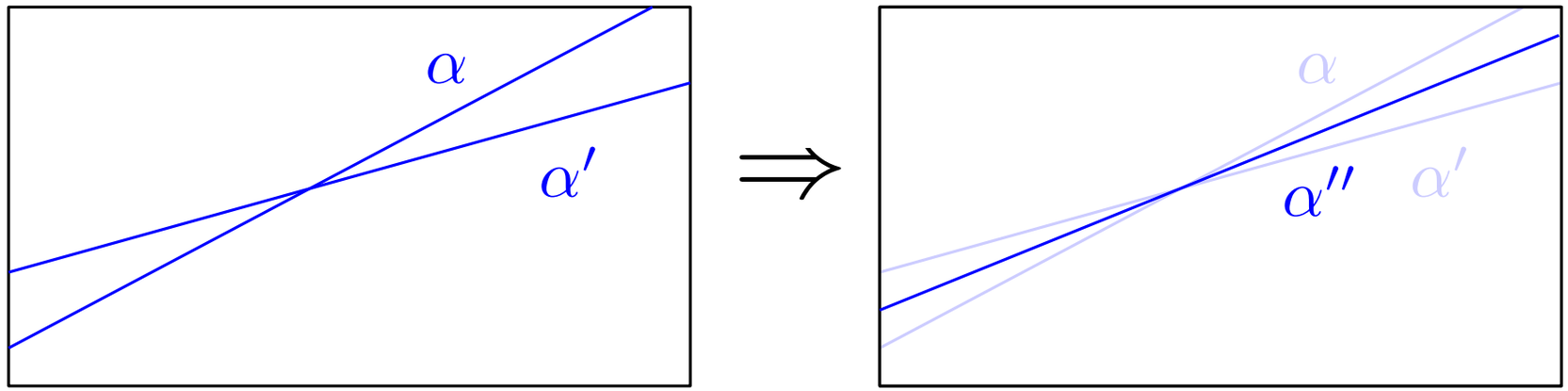
   $\forall e, e' \in E(G)$: perpindicular bisector of midpoints

   Axis score:

   $s_\alpha = 1$

   $s_\alpha =$ edge similarity

2. Group close axes



Edge-based symmetry [Klapauk *et al.* 2018]

Reflective symmetry:

1. Generate axes

   $\forall e \in E(G)$: perpindicular bisector

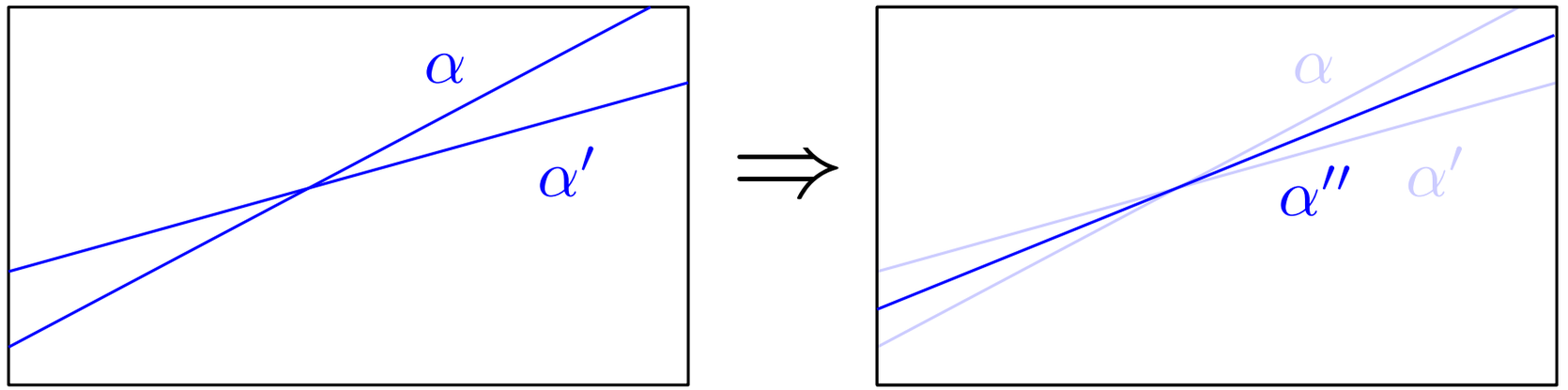
   $\forall e, e' \in E(G)$: perpindicular bisector of midpoints

   Axis score:

   $s_\alpha = 1$

   $s_\alpha =$ edge similarity

2. Group close axes



$s_{\alpha''} = s_\alpha + s_{\alpha'}$

Edge-based symmetry [Klapauk *et al.* 2018]

Reflective symmetry:

1. Generate axes

   $\forall e \in E(G)$: perpindicular bisector

   $\forall e, e' \in E(G)$: perpindicular bisector of midpoints

2. Group close axes
3. Pick top N axes

Axis score:

$s_\alpha = 1$

$s_\alpha =$ edge similarity

Edge-based symmetry [Klapauk *et al.* 2018]

Reflective symmetry:

1. Generate axes

   $\forall e \in E(G)$: perpindicular bisector

   $\forall e, e' \in E(G)$: perpindicular bisector of midpoints

   Axis score:

   $s_\alpha = 1$

   $s_\alpha =$ edge similarity

2. Group close axes
3. Pick top N axes
4. Final score:

$$s_{\text{ref}} = \frac{\sum_\alpha^N \text{number of edges that voted for } \alpha}{N \cdot \text{number of edges}}$$

Edge-based symmetry [Klapauk *et al.* 2018]

Reflective symmetry:

1. Generate axes

   $\forall e \in E(G)$: perpindicular bisector

   $\forall e, e' \in E(G)$: perpindicular bisector of midpoints

   Axis score:

   $s_\alpha = 1$

   $s_\alpha =$ edge similarity

2. Group close axes
3. Pick top N axes
4. Final score:

$$s_{\text{ref}} = \frac{\sum_\alpha^N \text{number of edges that voted for } \alpha}{N \cdot \text{number of edges}}$$

- Similar for rotational and translational symmetry

Symmetry

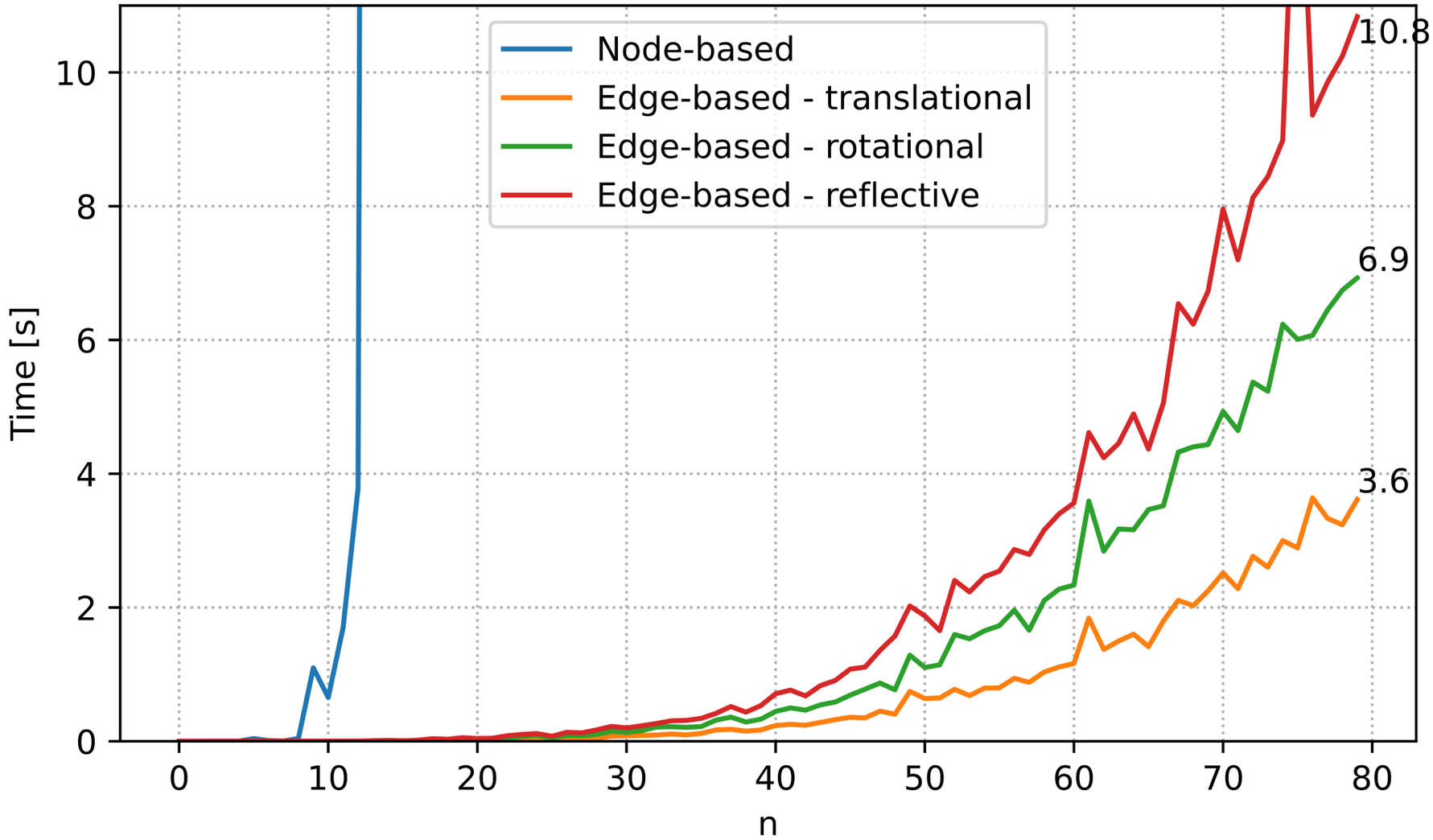
	Type	Runtime
Node-based	Reflective	$O(n^7)$
Edge-based	Reflective, rotational, translational	$O(m^2 + N)$
Stress-based	n.a.	$O(n \log n)$
Force-based	n.a.	$O(n \log n)$
Machine Learning	Reflective	$O(n + m)$
Automorphism detection	Reflective, translational	$O(n \log n)$
Visual symmetry	Reflective, rotational, translational	$O(n + m)$

Symmetry

	Type	**Runtime**
Node-based	Reflective	$O(n^7)$
Edge-based	Reflective, rotational, translational	$O(m^2 + N)$
Stress-based	n.a.	$O(n \log n)$
Force-based	n.a.	$O(n \log n)$
Machine Learning	Reflective	$O(n + m)$
Automorphism detection	Reflective, translational	$O(n \log n)$
Visual symmetry	Reflective, rotational, translational	$O(n + m)$

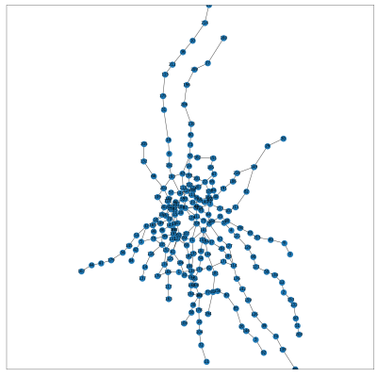
Visual Symmetry

- Random graphs with edge density 0.25



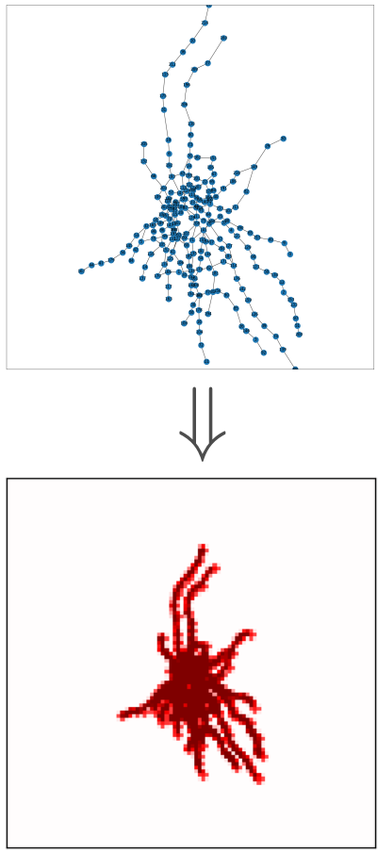
Visual Symmetry

Subway of Tokyo



Visual Symmetry

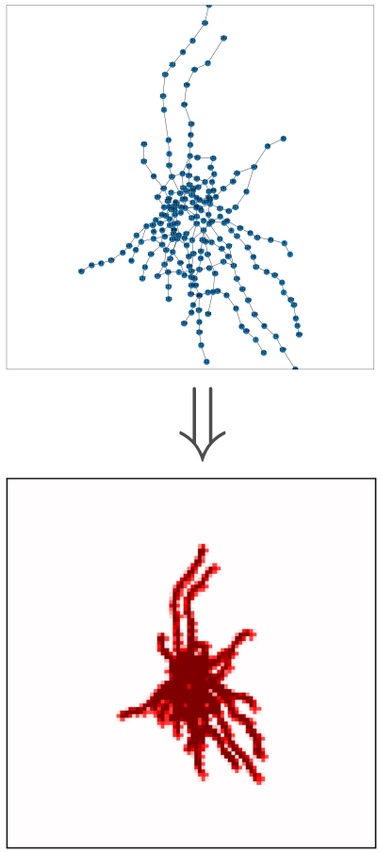
Subway of Tokyo



1. Draw pixel image

Visual Symmetry

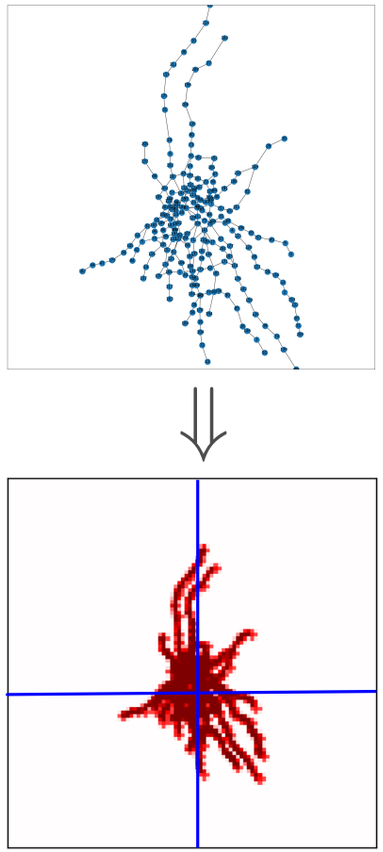
Subway of Tokyo



1. Draw pixel image
2. Translate to centroid

Visual Symmetry

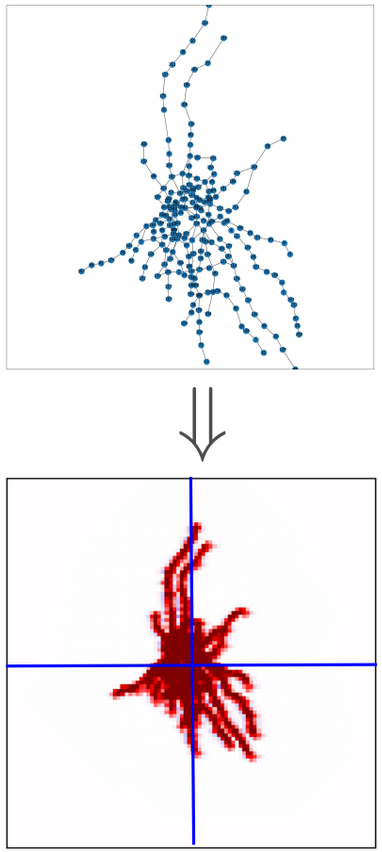
Subway of Tokyo



1. Draw pixel image
2. Translate to centroid

Visual Symmetry

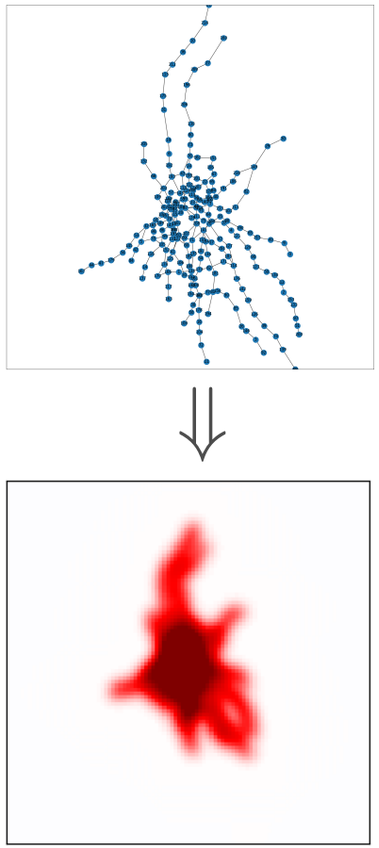
Subway of Tokyo



1. Draw pixel image
2. Translate to centroid

Visual Symmetry

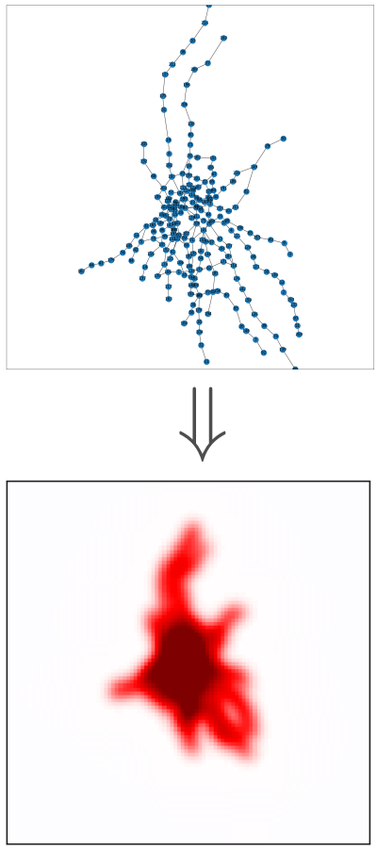
Subway of Tokyo



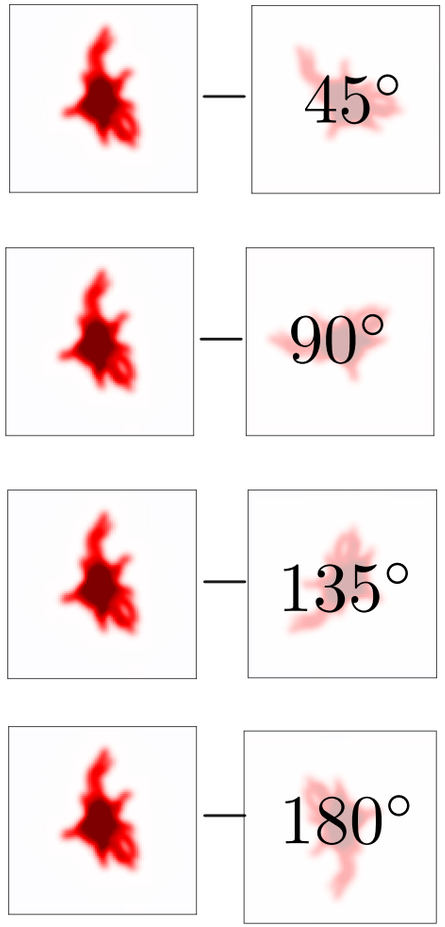
1. Draw pixel image
2. Translate to centroid
3. Gaussian Blur

Visual Symmetry

Subway of Tokyo

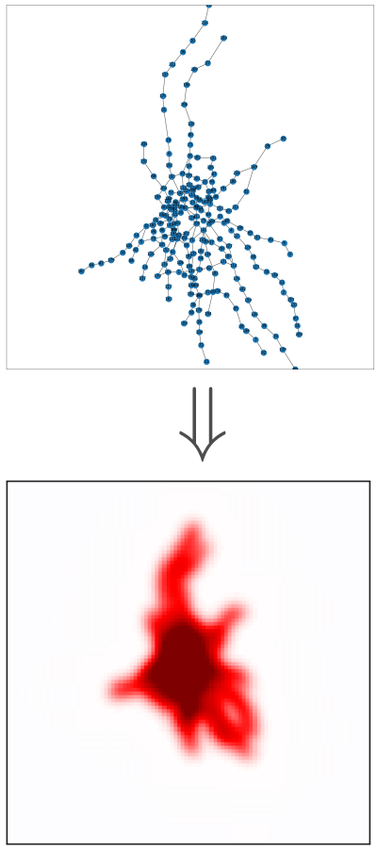


Rotational symmetry:

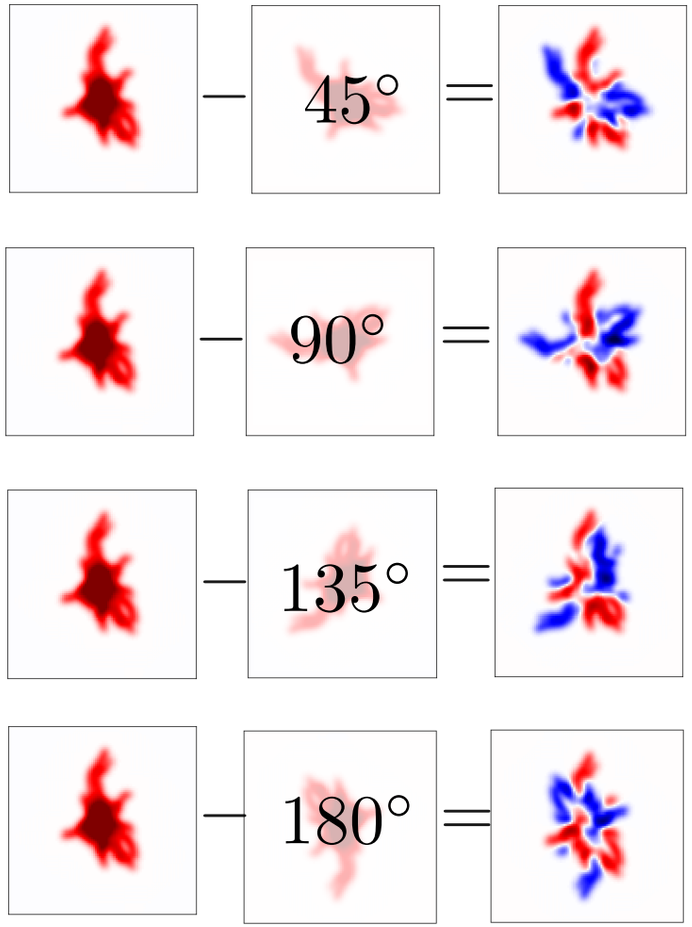


Visual Symmetry

Subway of Tokyo

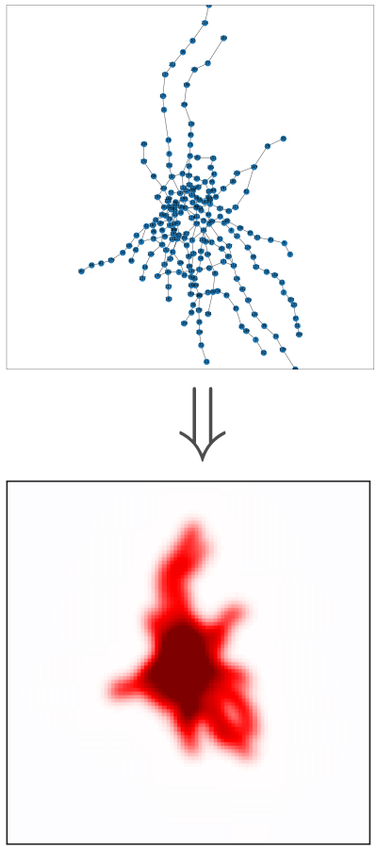


Rotational symmetry:

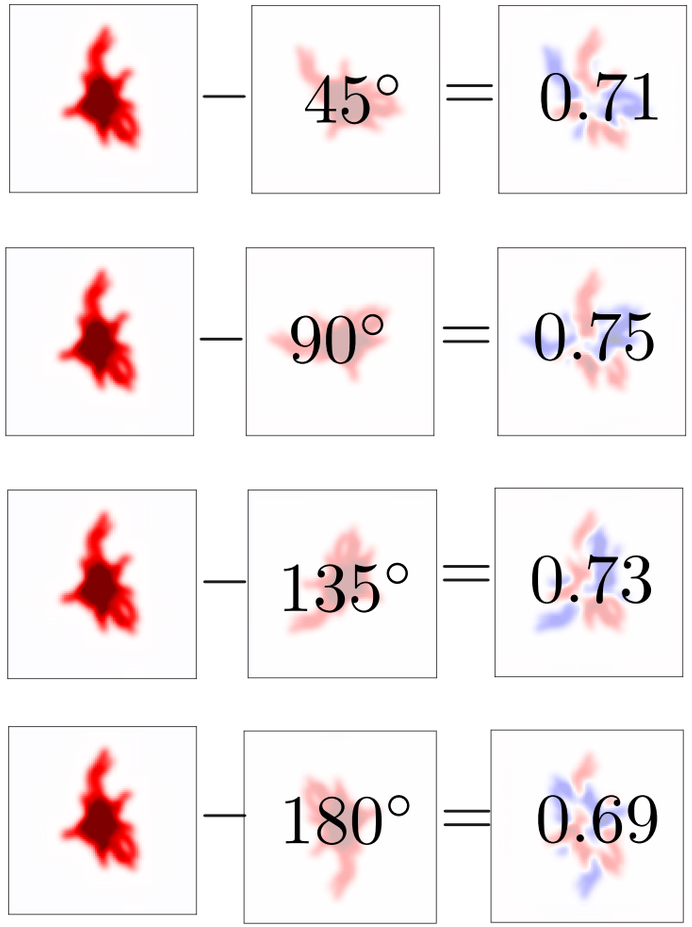


Visual Symmetry

Subway of Tokyo

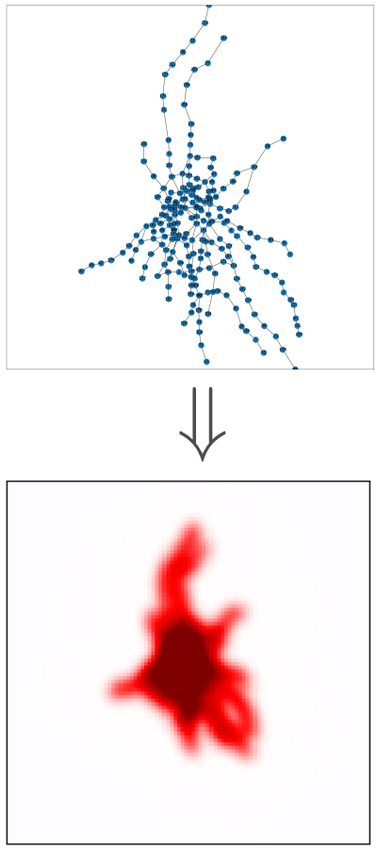


Rotational symmetry:

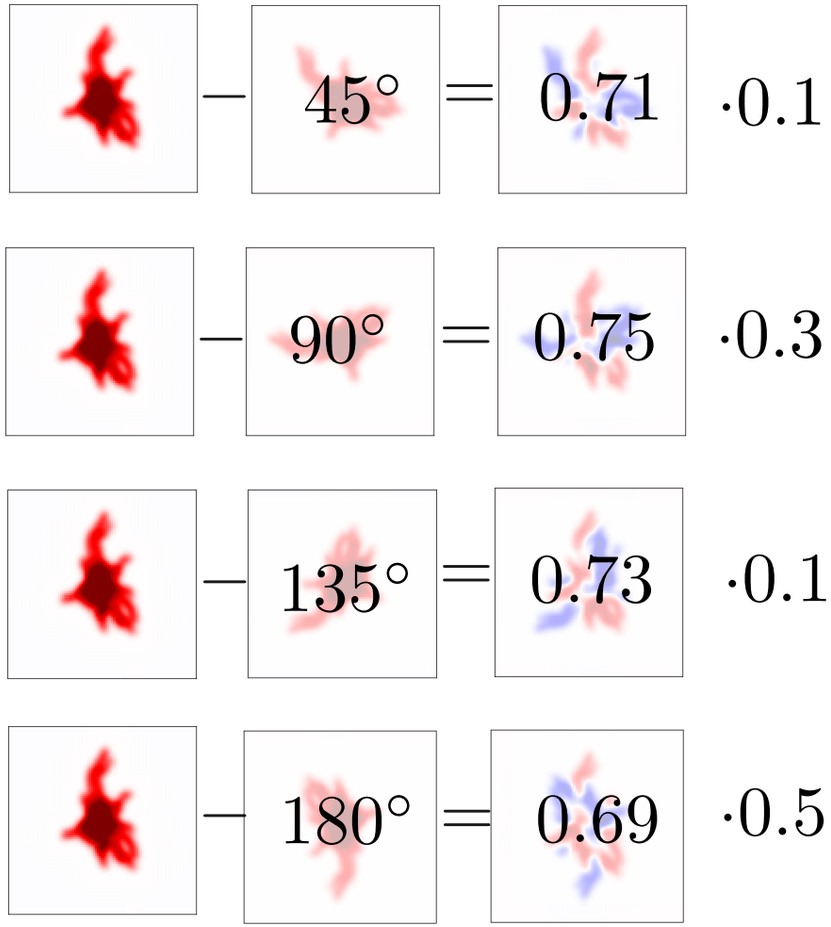


Visual Symmetry

Subway of Tokyo

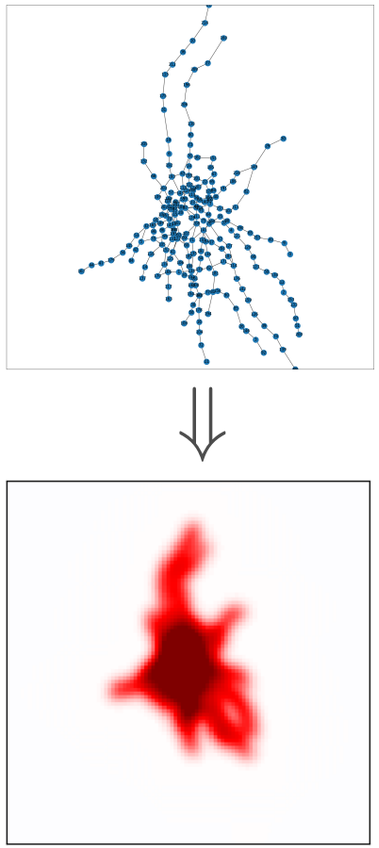


Rotational symmetry:

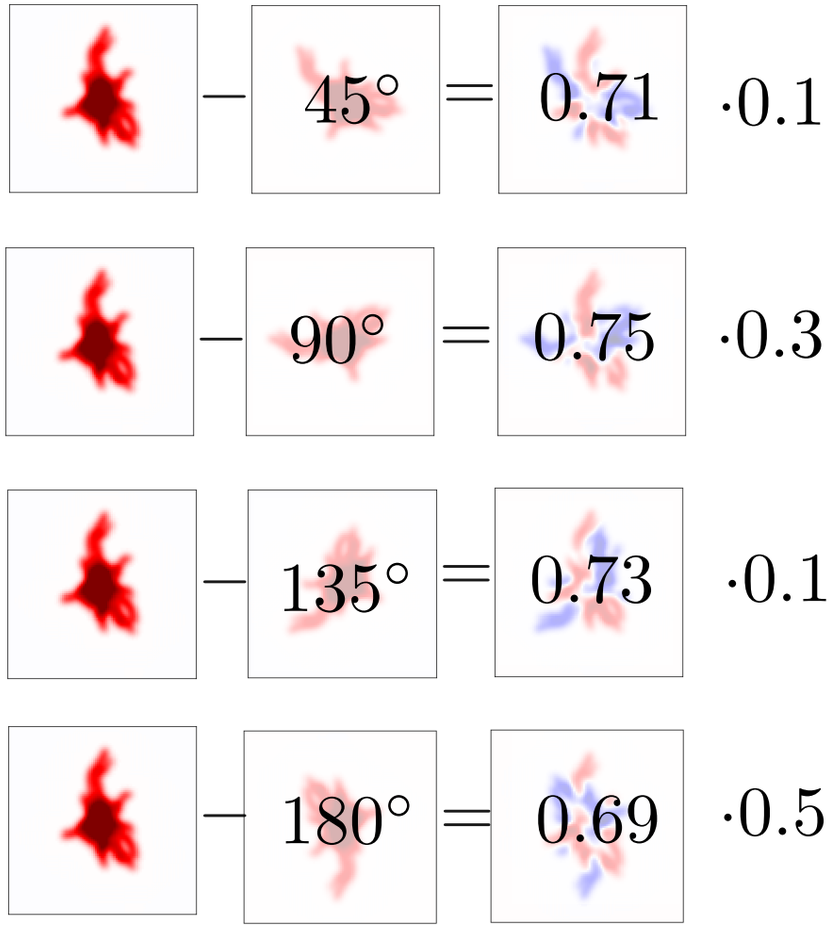


Visual Symmetry

Subway of Tokyo



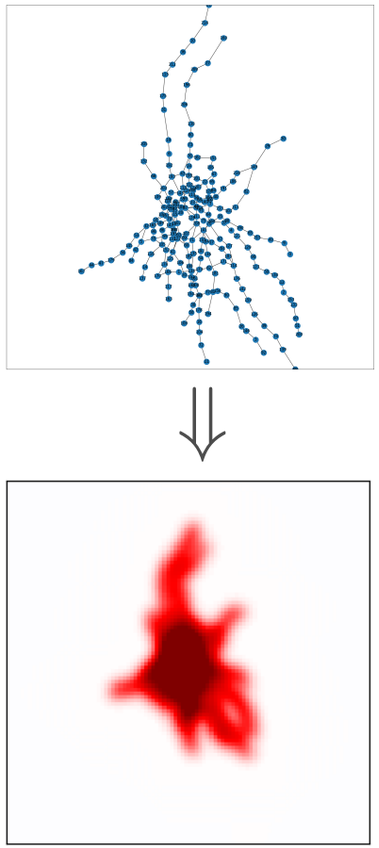
Rotational symmetry:



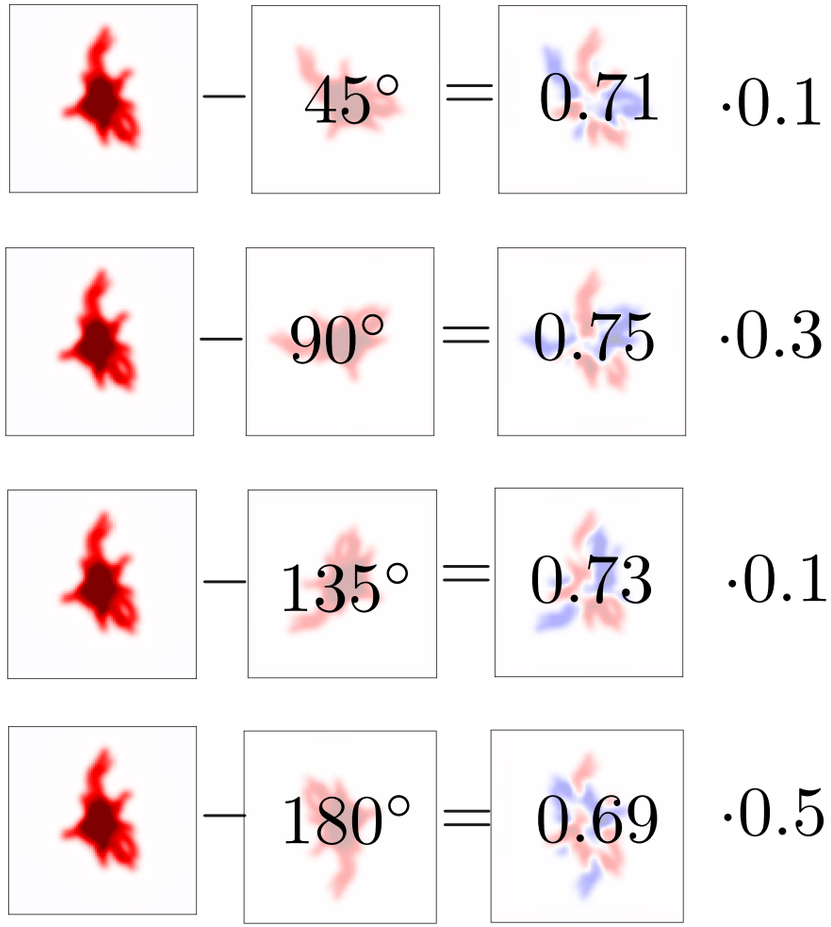
$$\bar{x}_w = 0.78$$

Visual Symmetry

Subway of Tokyo



Rotational symmetry:

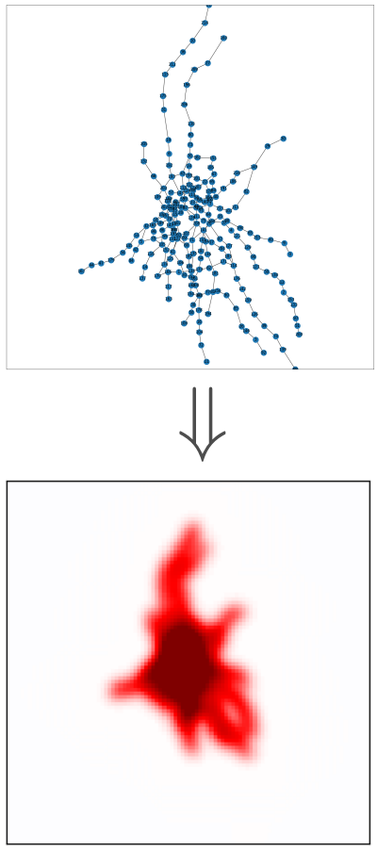


$$\bar{x}_w = 0.78$$

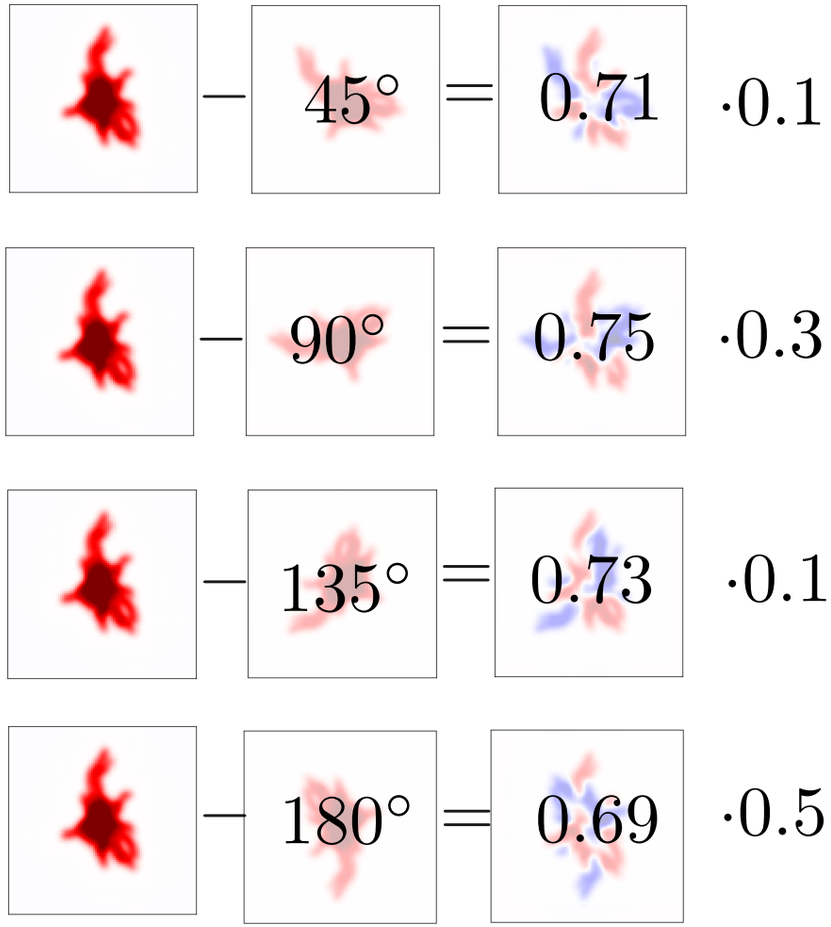
$$1 - \bar{x}_w^2 = 0.39$$

Visual Symmetry

Subway of Tokyo



Rotational symmetry:



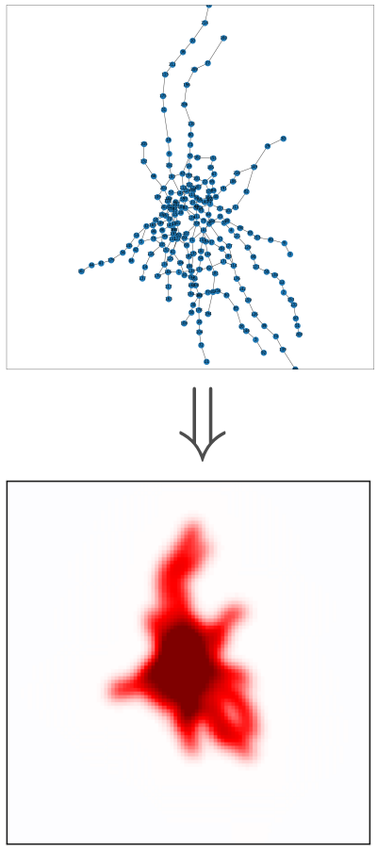
$$\bar{x}_w = 0.78$$

$$1 - \bar{x}_w^2 = 0.39$$

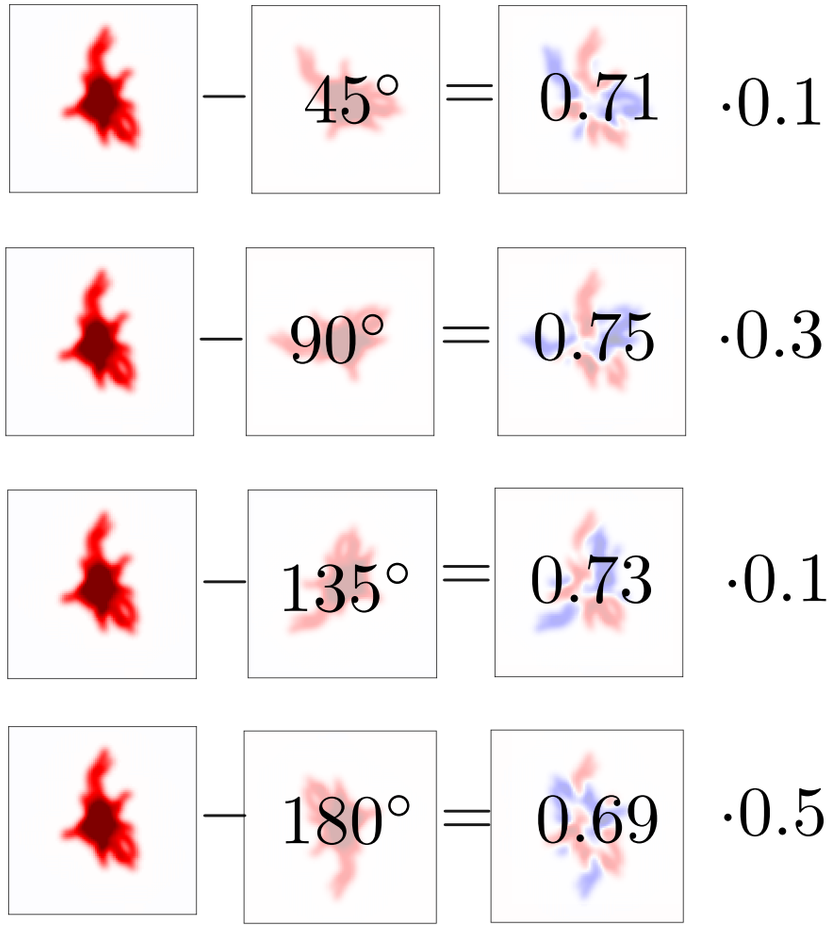
Reflective symmetry:

Visual Symmetry

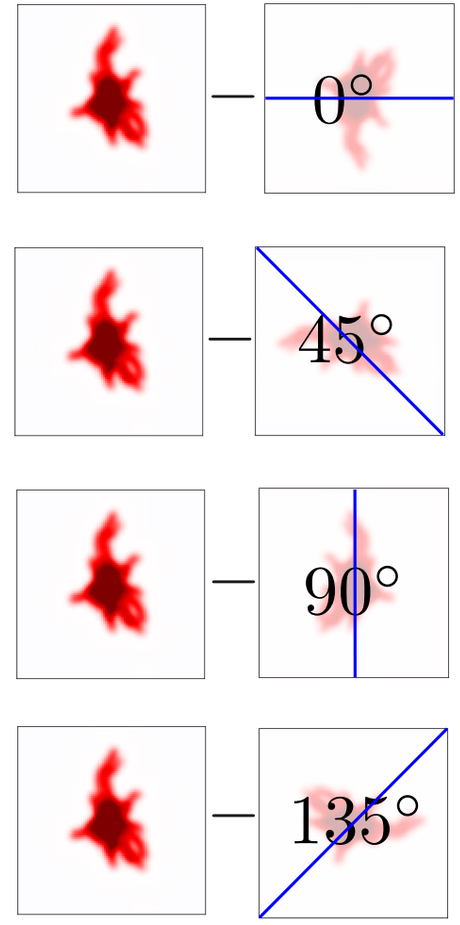
Subway of Tokyo



Rotational symmetry:



Reflective symmetry:

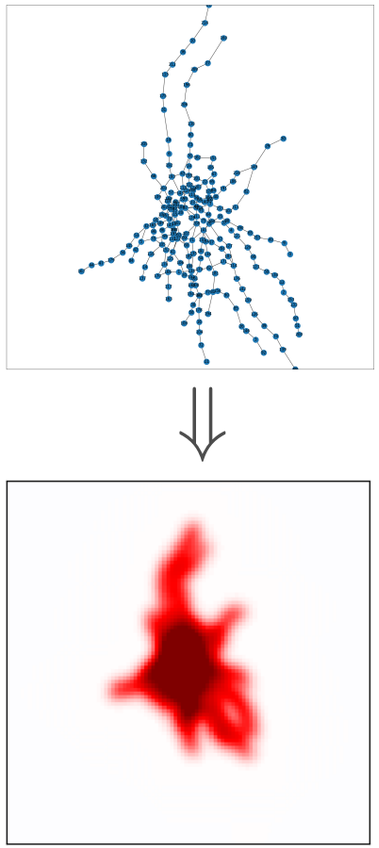


$$\bar{x}_w = 0.78$$

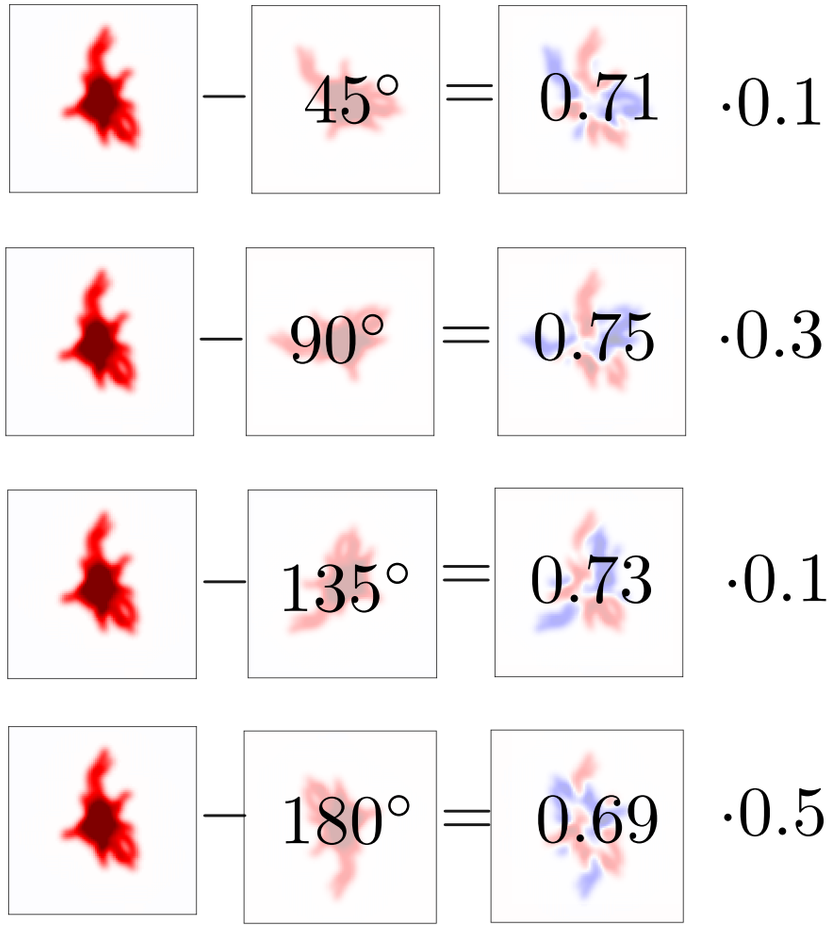
$$1 - \bar{x}_w^2 = 0.39$$

Visual Symmetry

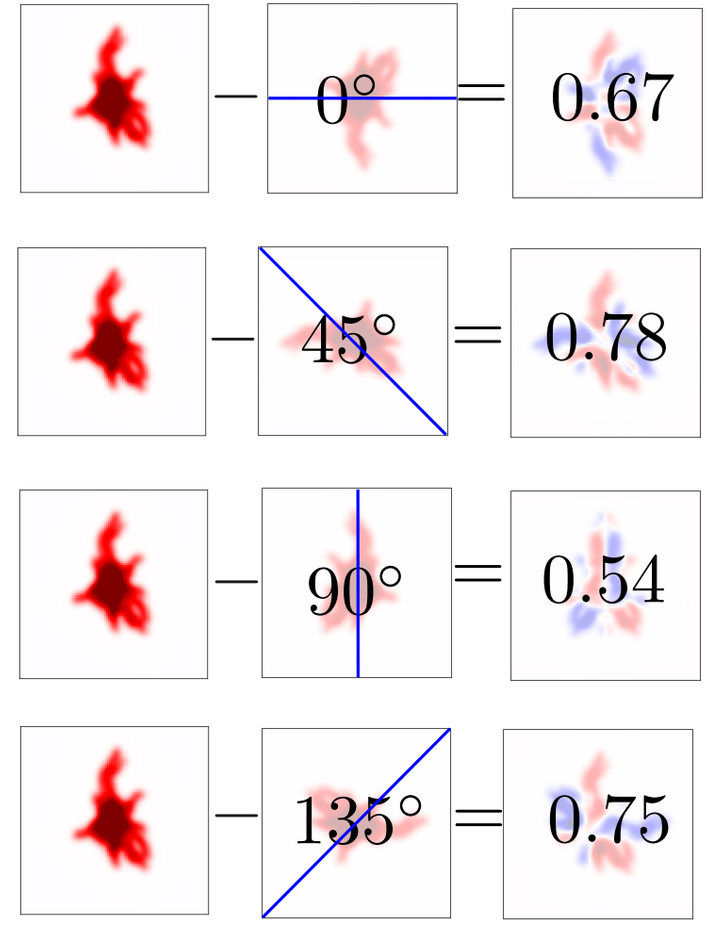
Subway of Tokyo



Rotational symmetry:



Reflective symmetry:

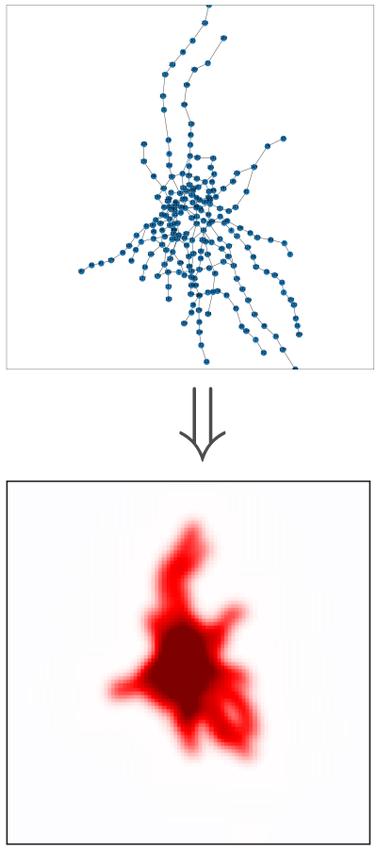


$$\bar{x}_w = 0.78$$

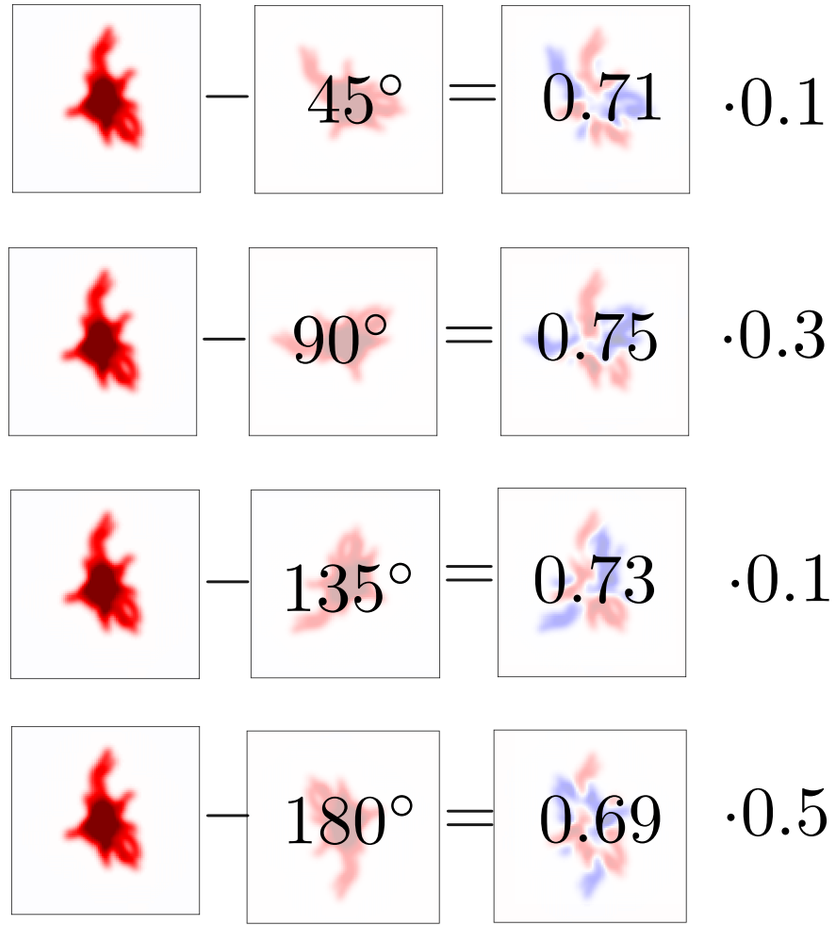
$$1 - \bar{x}_w^2 = 0.39$$

Visual Symmetry

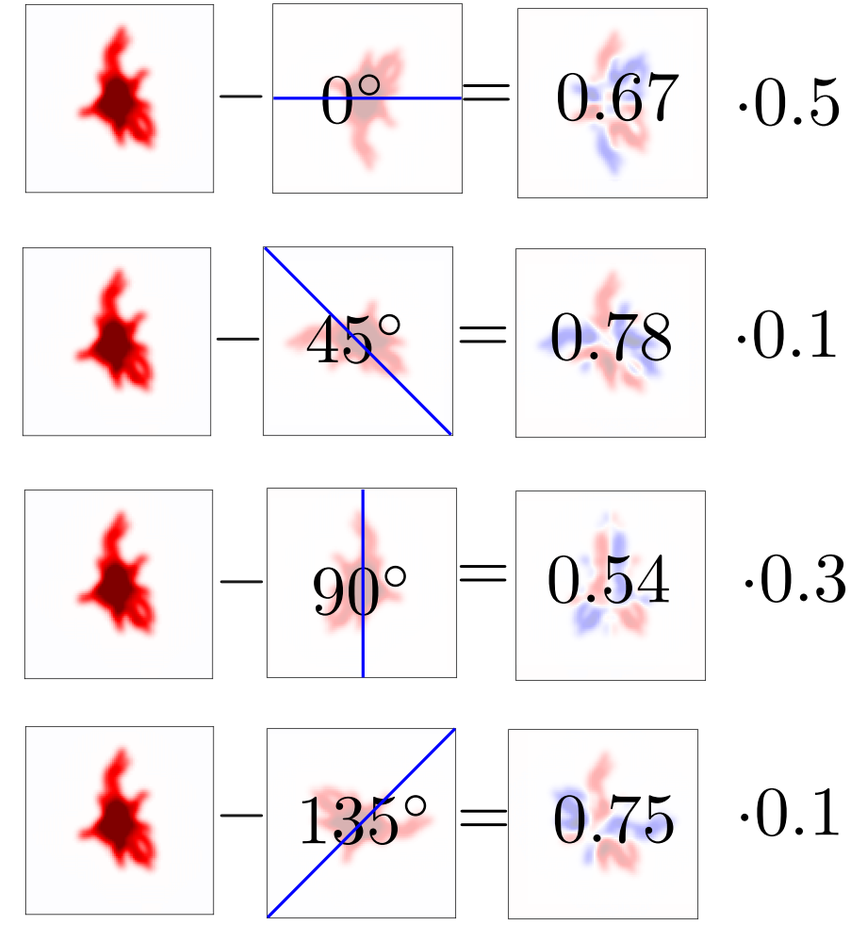
Subway of Tokyo



Rotational symmetry:



Reflective symmetry:

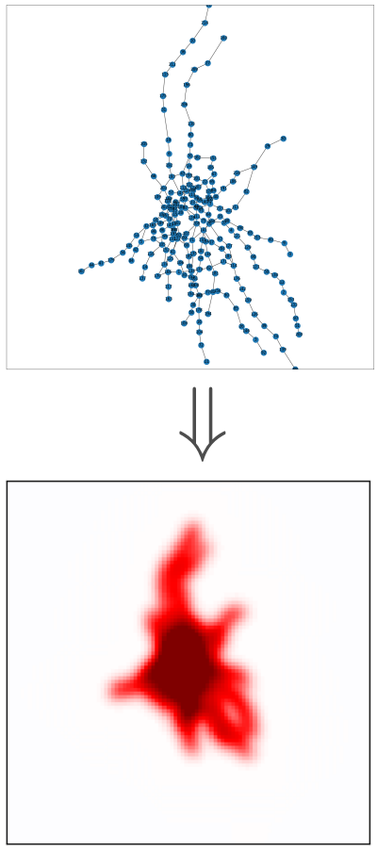


$$\bar{x}_w = 0.78$$

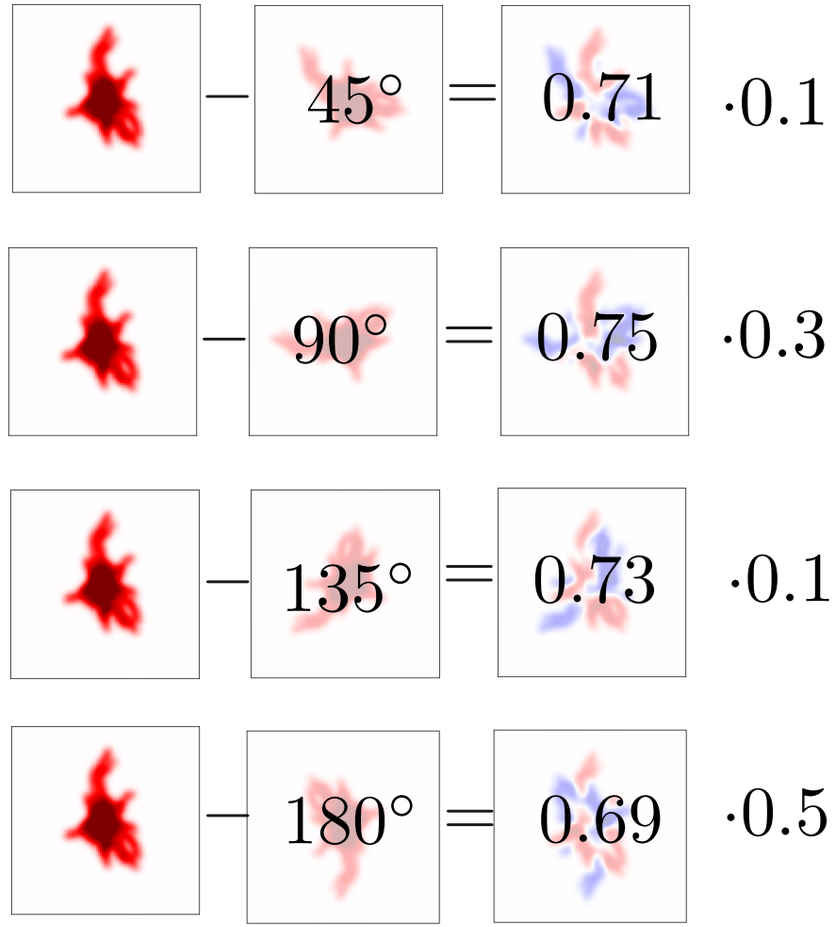
$$1 - \bar{x}_w^2 = 0.39$$

Visual Symmetry

Subway of Tokyo



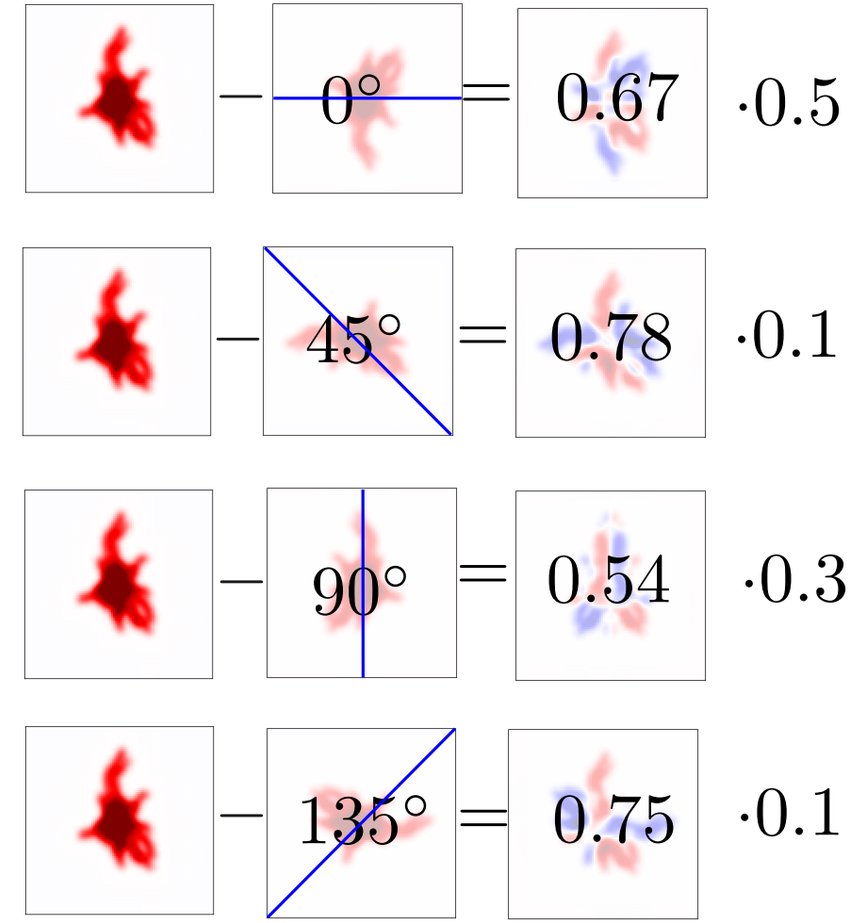
Rotational symmetry:



$$\bar{x}_w = 0.78$$

$$1 - \bar{x}_w^2 = 0.39$$

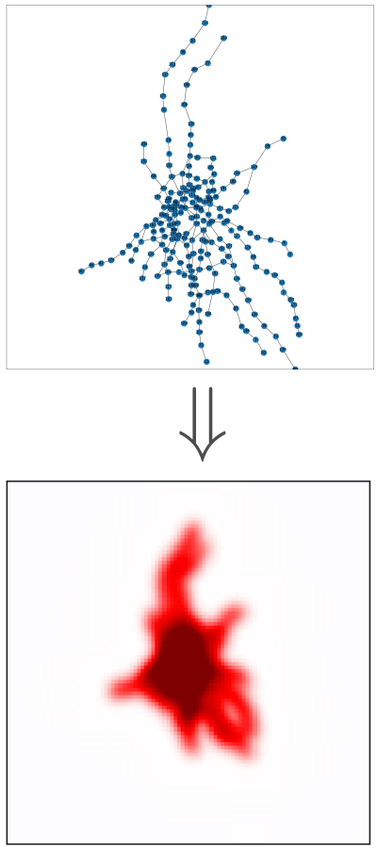
Reflective symmetry:



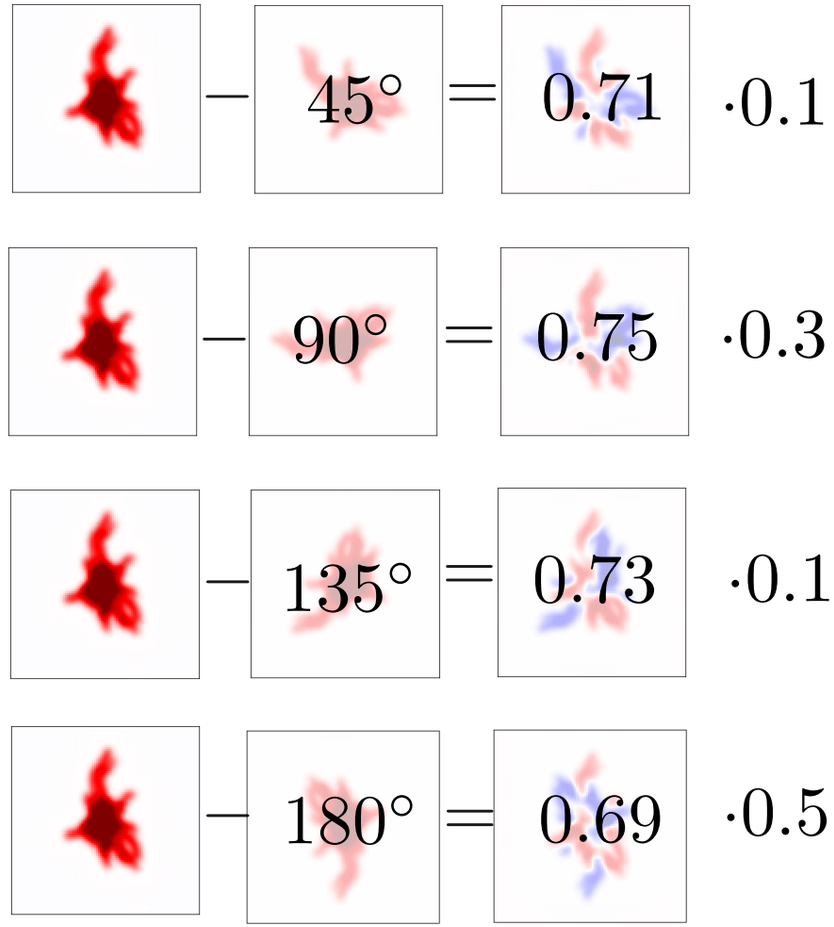
$$\bar{x}_w = 0.65$$

Visual Symmetry

Subway of Tokyo



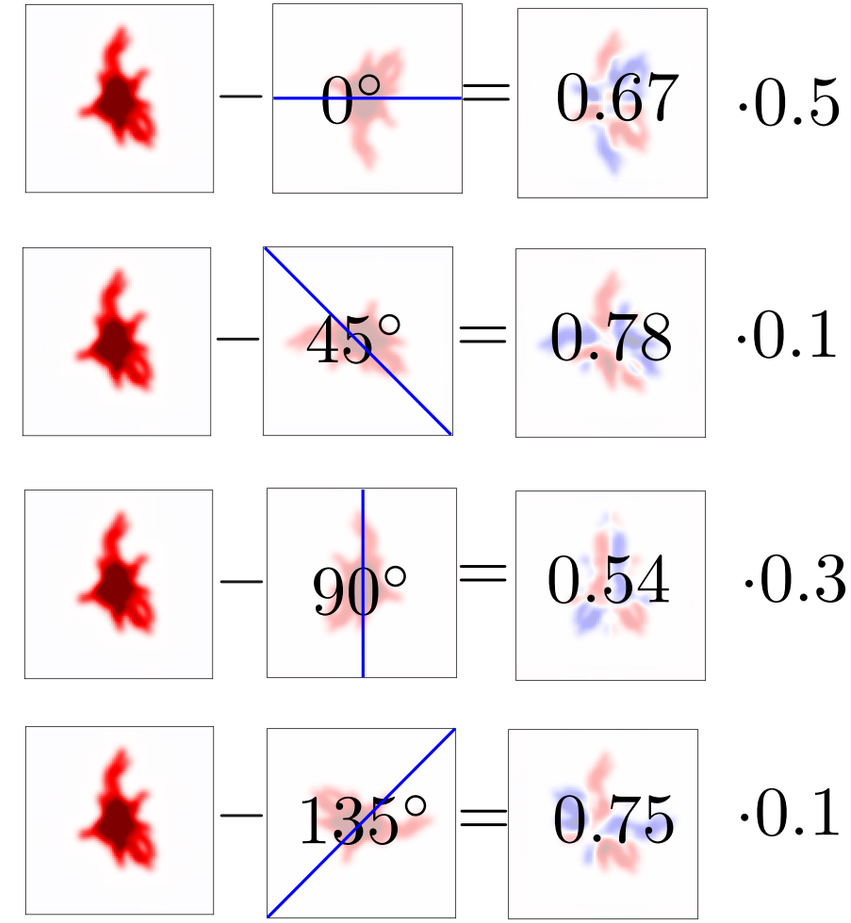
Rotational symmetry:



$$\bar{x}_w = 0.78$$

$$1 - \bar{x}_w^2 = 0.39$$

Reflective symmetry:

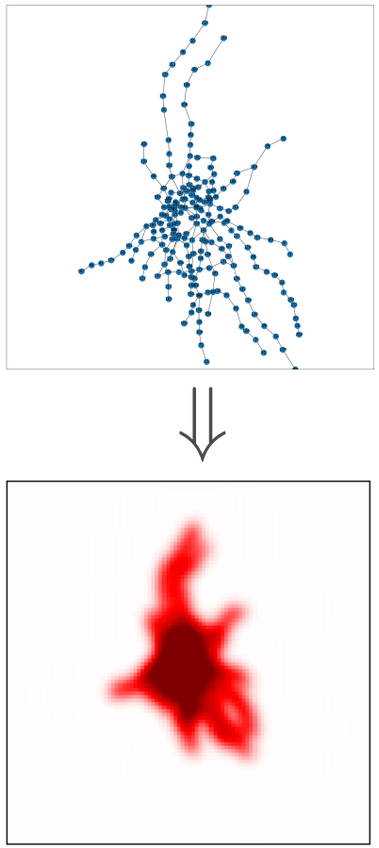


$$\bar{x}_w = 0.65$$

$$1 - \bar{x}_w^2 = 0.57$$

Visual Symmetry

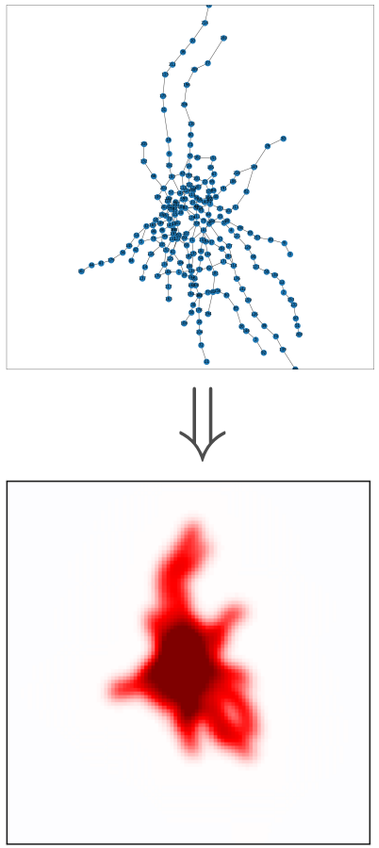
Subway of Tokyo



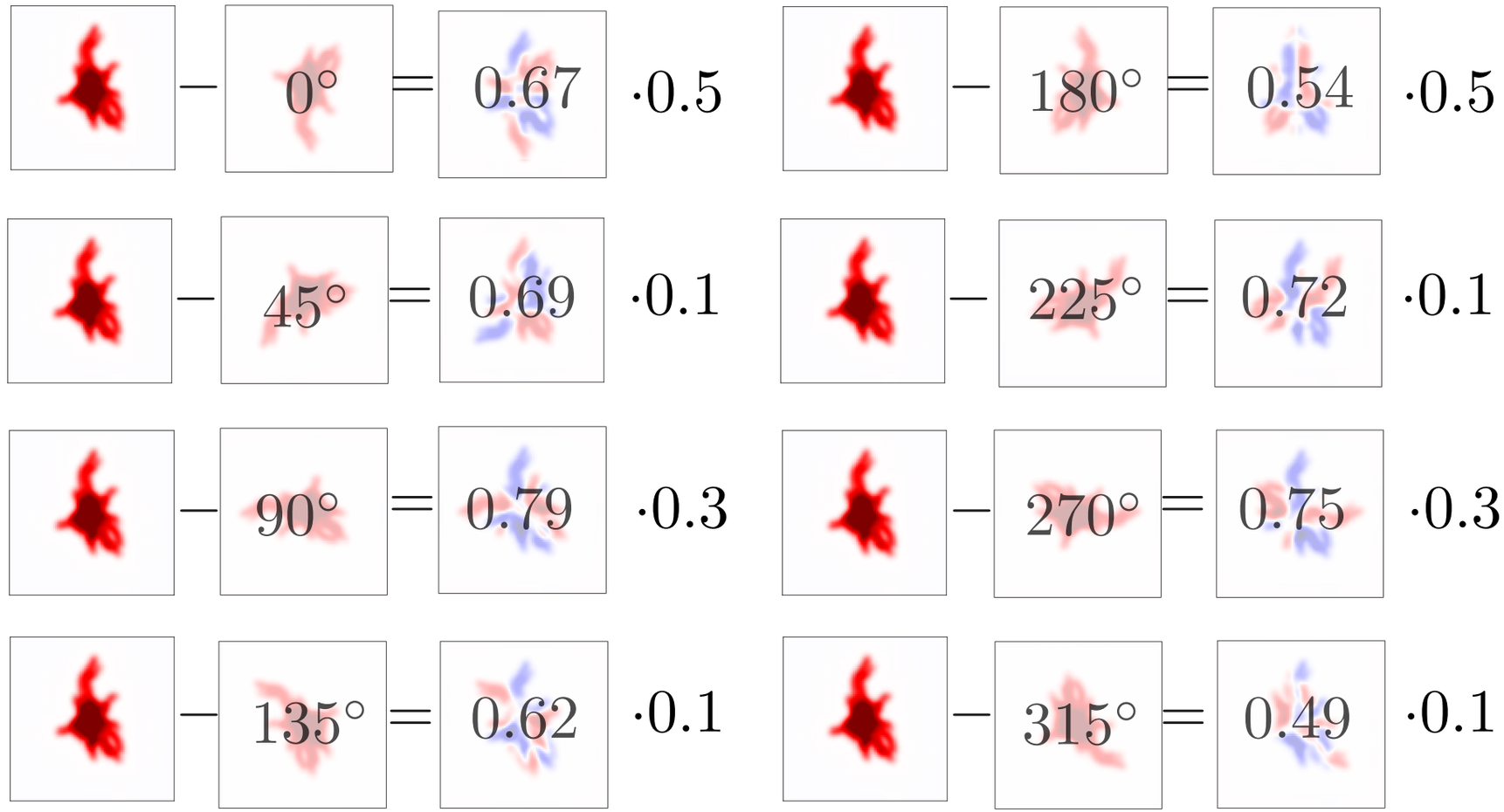
Dihedral symmetry:

Visual Symmetry

Subway of Tokyo

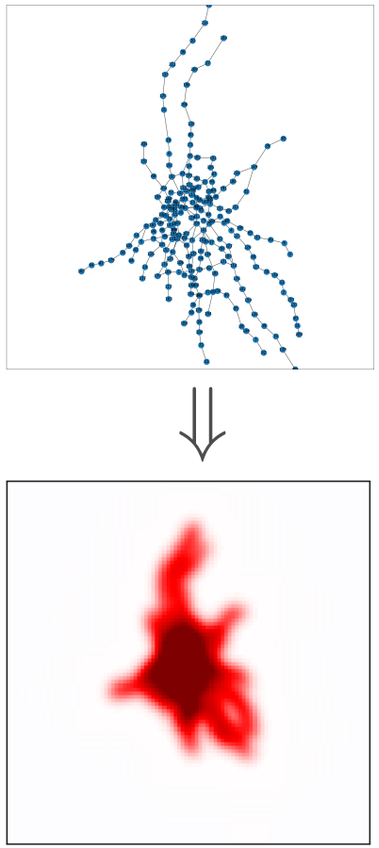


Dihedral symmetry:

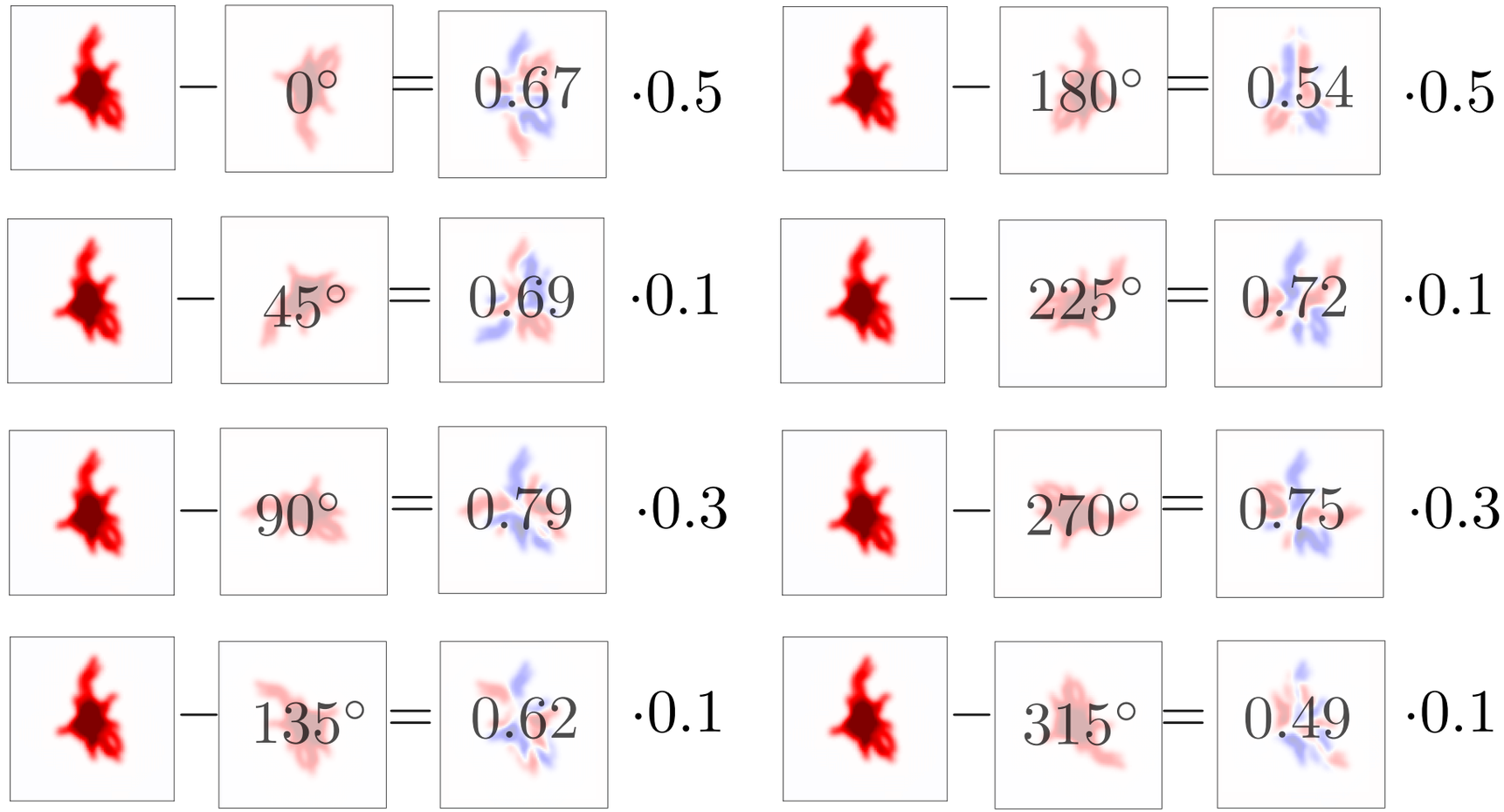


Visual Symmetry

Subway of Tokyo



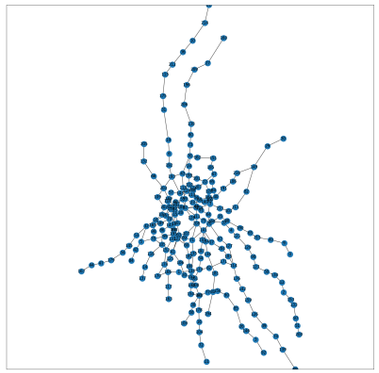
Dihedral symmetry:



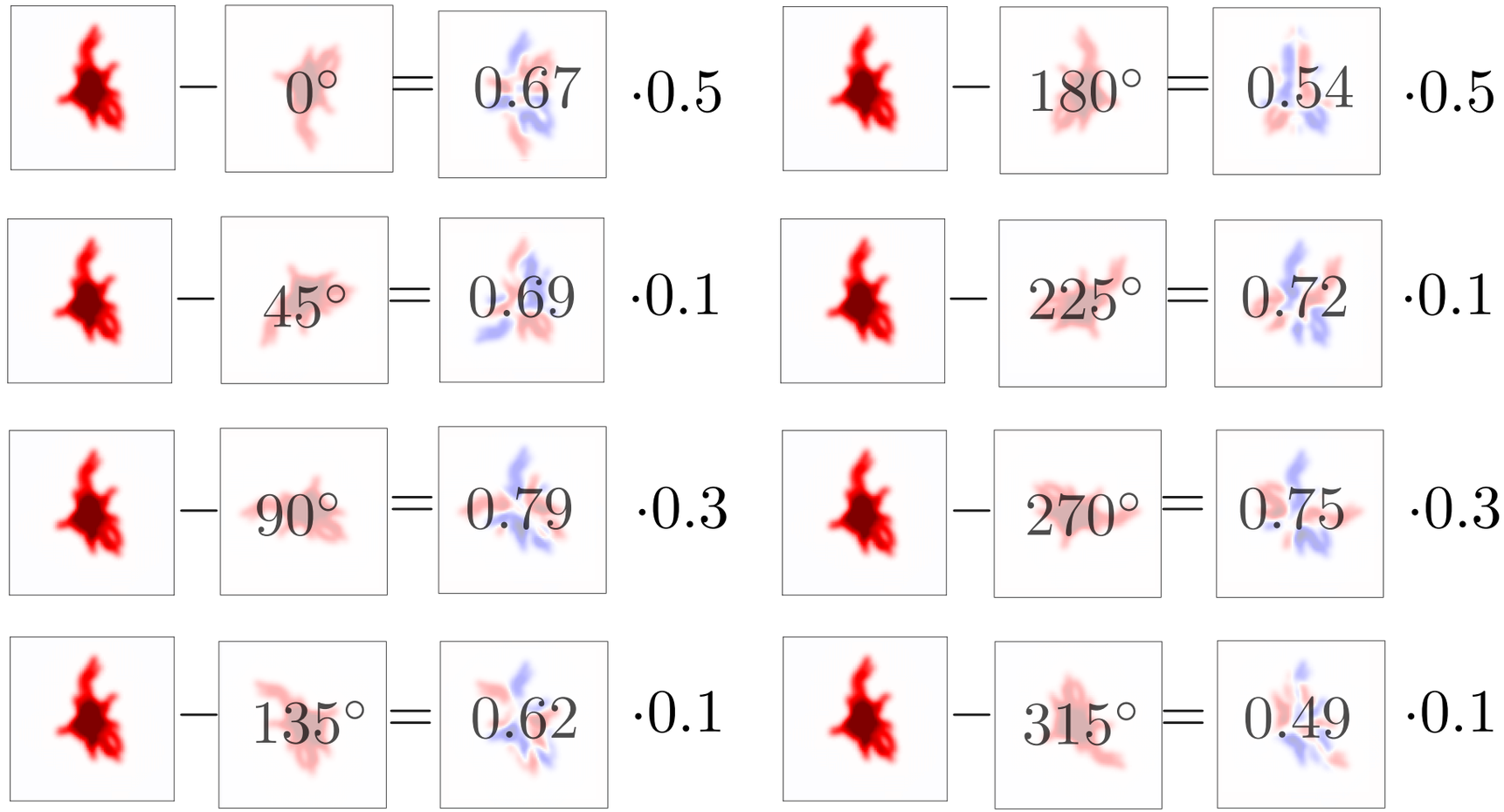
$$\bar{x}_w = 0.69$$

Visual Symmetry

Subway of Tokyo



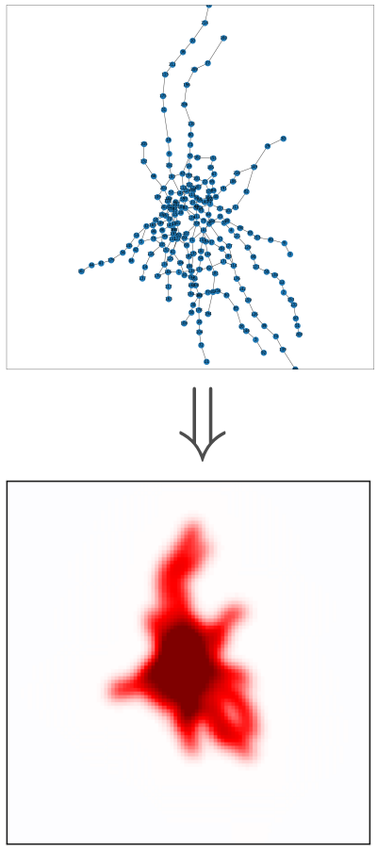
Dihedral symmetry:



$$\bar{x}_w = 0.69$$
$$1 - \bar{x}_w^2 = 0.52$$

Visual Symmetry

Subway of Tokyo



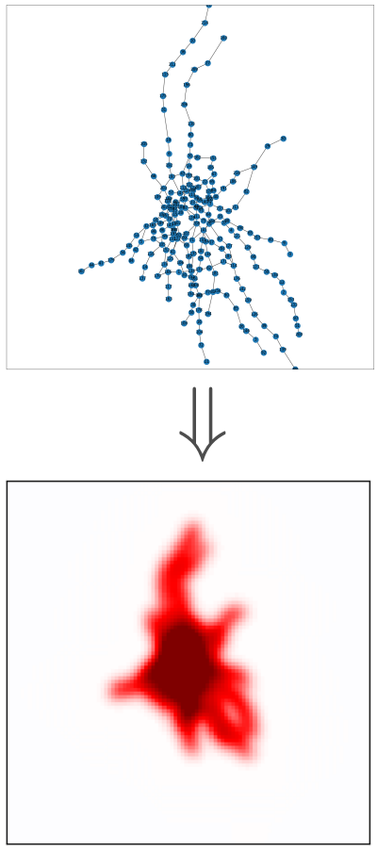
Rotational symmetry: 0.39

Reflective symmetry: 0.57

Dihedral symmetry: 0.52

Visual Symmetry

Subway of Tokyo



Rotational symmetry: 0.39

Reflective symmetry: 0.57

Dihedral symmetry: 0.52

Total: $\max(0.39, 0.57, 0.52) = 0.57$

Visual Symmetry

- Draw heatmap of graph into constant size matrix 

Visual Symmetry

- Draw heatmap of graph into constant size matrix 
- Center drawing and apply Gaussian blur

Visual Symmetry

- Draw heatmap of graph into constant size matrix ▦
- Center drawing and apply Gaussian blur
- Calculate metric for each type of symmetry

$$s_{\mathsf{rot}} = \frac{1}{\sum_\alpha s_\alpha} \sum_\alpha \frac{|▦ - \mathsf{rot}(▦, \alpha)|}{|x|} \cdot s_\alpha$$

$$s_{\mathsf{ref}} = \frac{1}{\sum_\alpha s_\alpha} \sum_\alpha \frac{|▦ - \mathsf{ref}(▦, \alpha)|}{|x|} \cdot s_\alpha$$

$$s_{\mathsf{dih}} = \frac{1}{\sum_\alpha s_\alpha} \sum_\alpha \frac{|▦ - \mathsf{dih}(▦, \alpha)|}{|x|} \cdot s_\alpha$$

Visual Symmetry

- Draw heatmap of graph into constant size matrix ▦
- Center drawing and apply Gaussian blur
- Calculate metric for each type of symmetry

Axis-specific factor

$$s_{\text{rot}} = \frac{1}{\sum_\alpha s_\alpha} \sum_\alpha \frac{|▦ - \text{rot}(▦, \alpha)|}{|x|} \cdot s_\alpha$$

$$s_{\text{ref}} = \frac{1}{\sum_\alpha s_\alpha} \sum_\alpha \frac{|▦ - \text{ref}(▦, \alpha)|}{|x|} \cdot s_\alpha$$

$$s_{\text{dih}} = \frac{1}{\sum_\alpha s_\alpha} \sum_\alpha \frac{|▦ - \text{dih}(▦, \alpha)|}{|x|} \cdot s_\alpha$$

Visual Symmetry

- Draw heatmap of graph into constant size matrix ▦
- Center drawing and apply Gaussian blur
- Calculate metric for each type of symmetry

Axis-specific factor
[De Luca *et al.* 2018]

$$s_{\text{rot}} = \frac{1}{\sum_\alpha s_\alpha} \sum_\alpha \frac{|▦ - \text{rot}(▦, \alpha)|}{|x|} \cdot s_\alpha$$

$$s_{\text{ref}} = \frac{1}{\sum_\alpha s_\alpha} \sum_\alpha \frac{|▦ - \text{ref}(▦, \alpha)|}{|x|} \cdot s_\alpha$$

$$s_{\text{dih}} = \frac{1}{\sum_\alpha s_\alpha} \sum_\alpha \frac{|▦ - \text{dih}(▦, \alpha)|}{|x|} \cdot s_\alpha$$

Visual Symmetry

- Draw heatmap of graph into constant size matrix ▦
- Center drawing and apply Gaussian blur
- Calculate metric for each type of symmetry

Axis-specific factor
[De Luca *et al.* 2018]

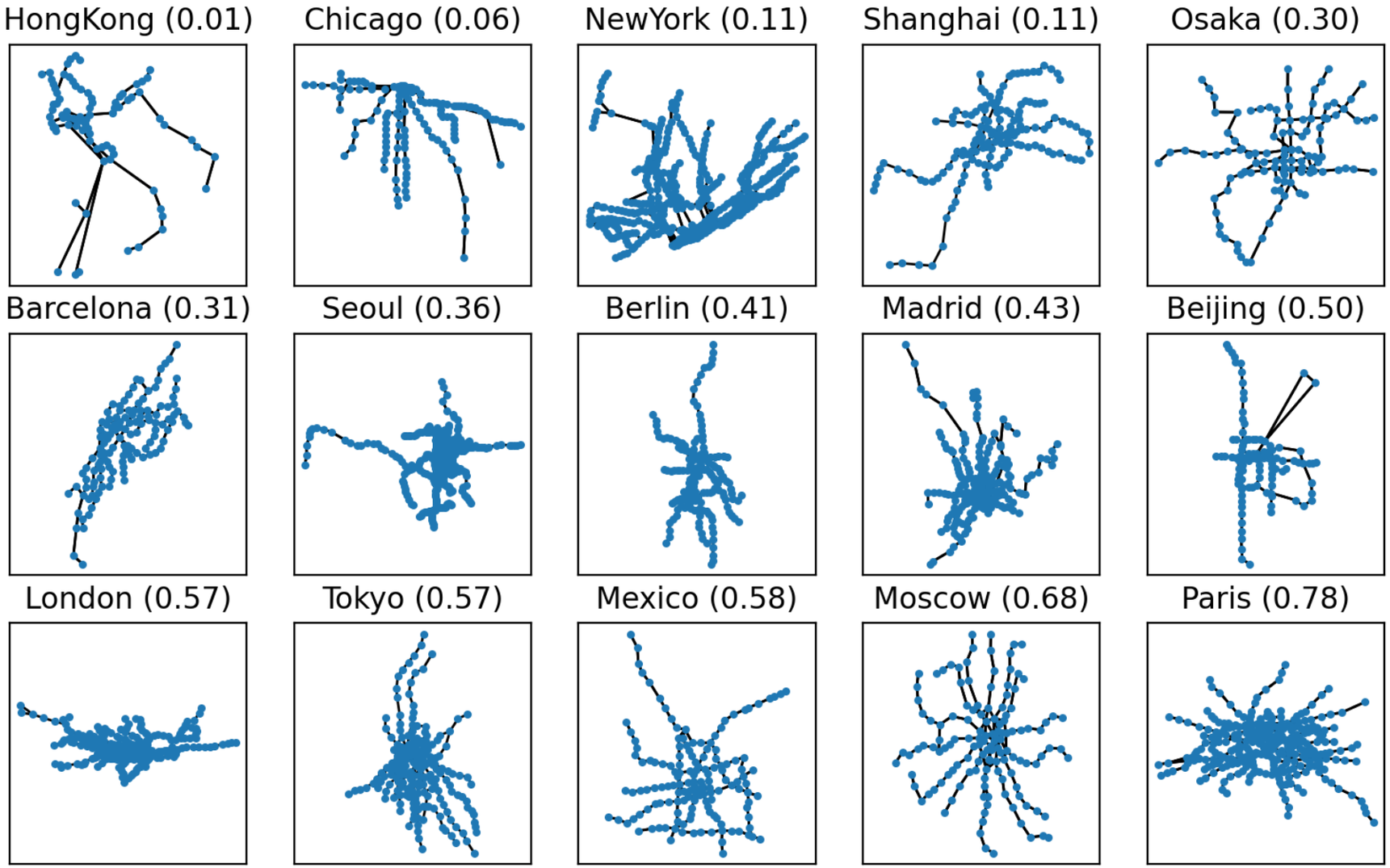
$$s_{\mathsf{rot}} = \frac{1}{\sum_\alpha s_\alpha} \sum_\alpha \frac{|▦ - \mathsf{rot}(▦, \alpha)|}{|x|} \cdot s_\alpha$$

$$s_{\mathsf{ref}} = \frac{1}{\sum_\alpha s_\alpha} \sum_\alpha \frac{|▦ - \mathsf{ref}(▦, \alpha)|}{|x|} \cdot s_\alpha$$

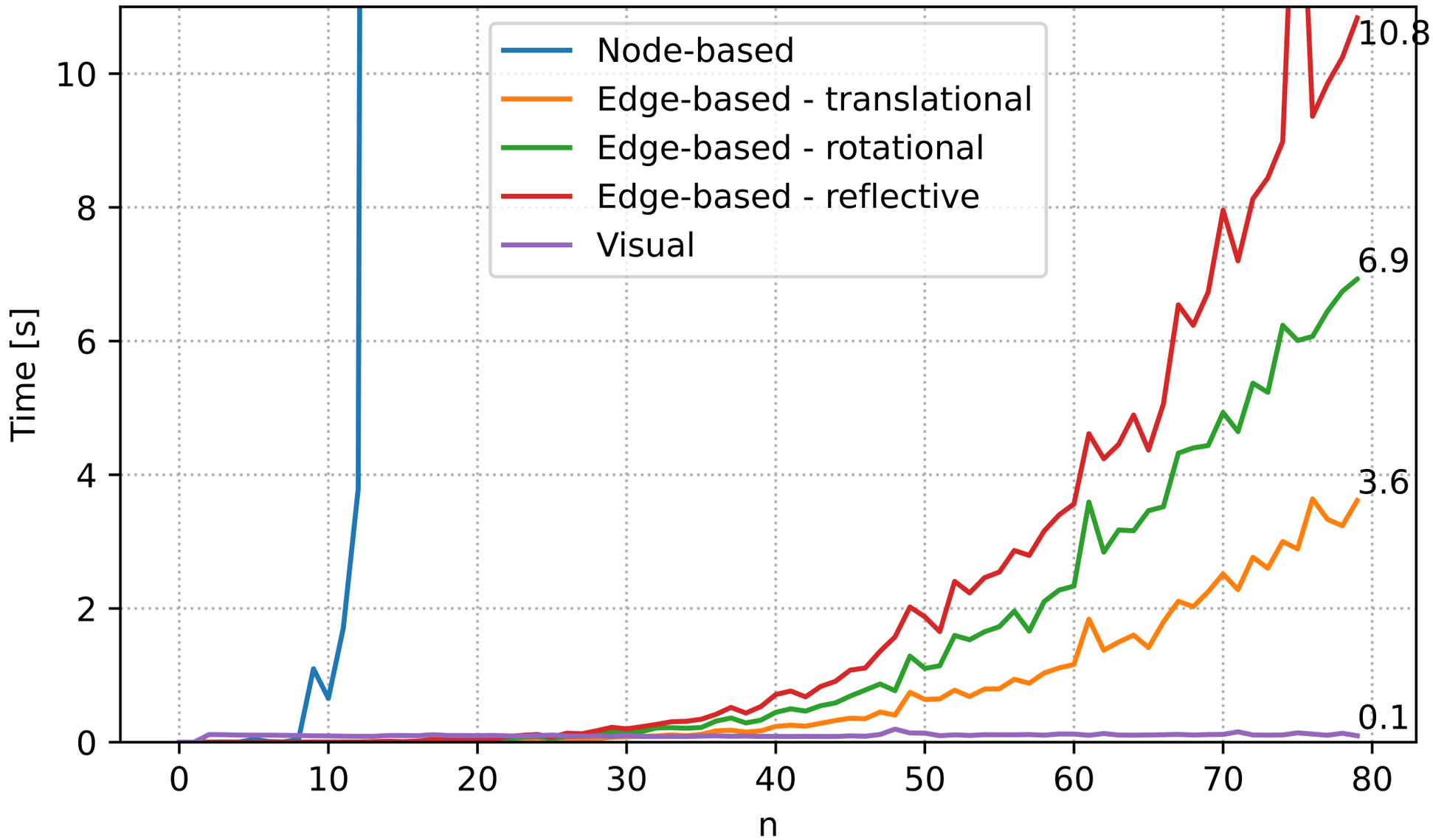
$$s_{\mathsf{dih}} = \frac{1}{\sum_\alpha s_\alpha} \sum_\alpha \frac{|▦ - \mathsf{dih}(▦, \alpha)|}{|x|} \cdot s_\alpha$$

- Final metric: $1 - \min(s_{\mathsf{rot}}, s_{\mathsf{ref}}, s_{\mathsf{dih}})^2$

Symmetry of Subways



Runtime

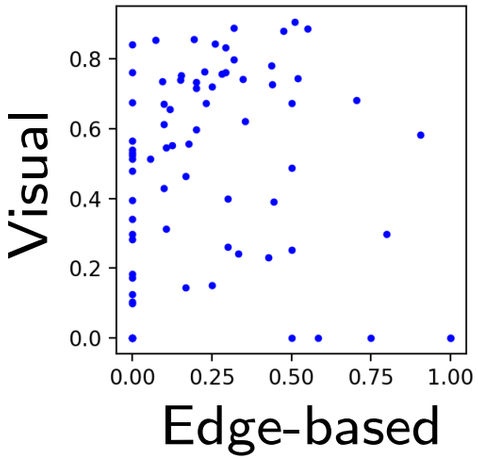

Node-based
Edge-based - translational
Edge-based - rotational
Edge-based - reflective
Visual
Time [s]
n
0
2
4
6
8
10
0
10
20
30
40
50
60
70
80
10.8
6.9
3.6
0.1

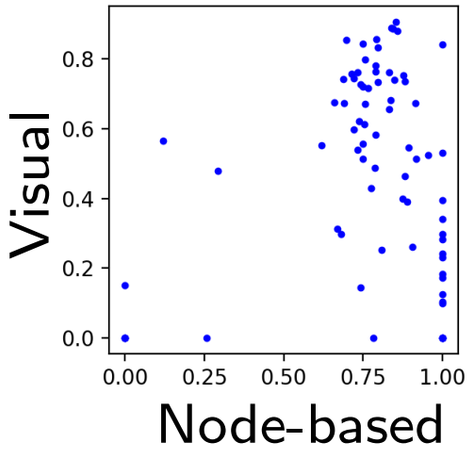
Problems

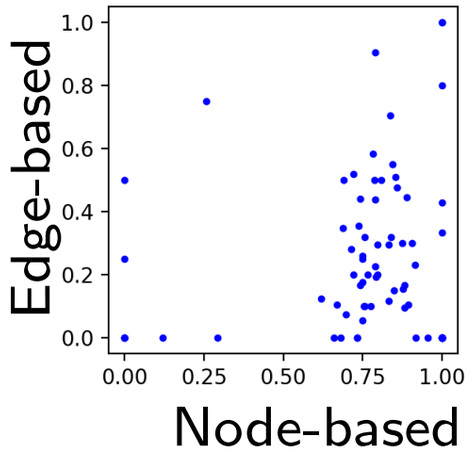
- Weak correlation

Problems

- Weak correlation

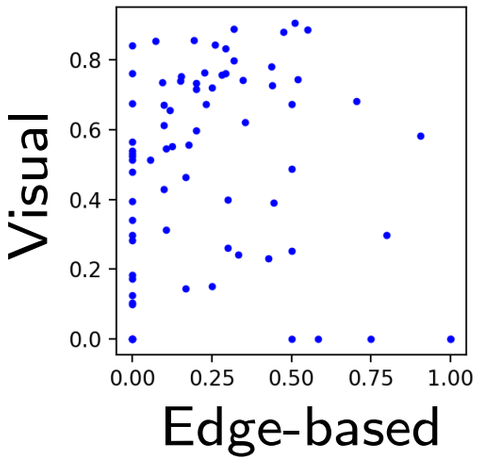


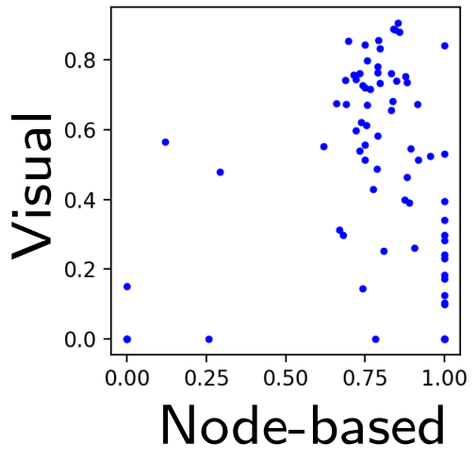


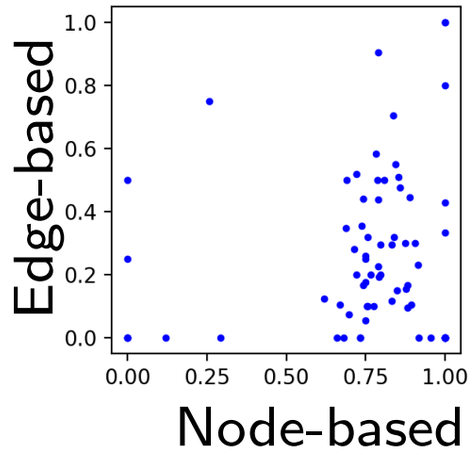


Problems

- Weak correlation





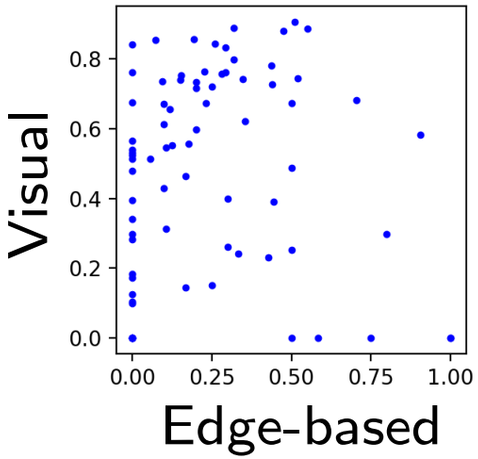


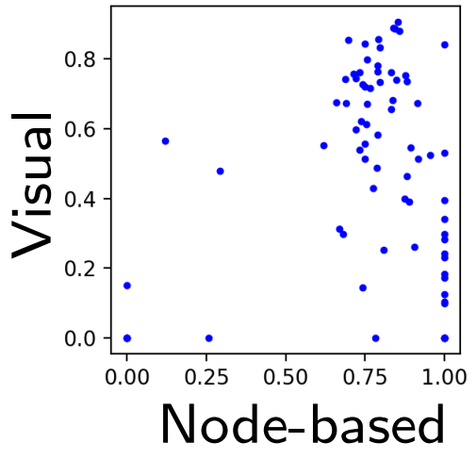
70 random graphs of size 5-12

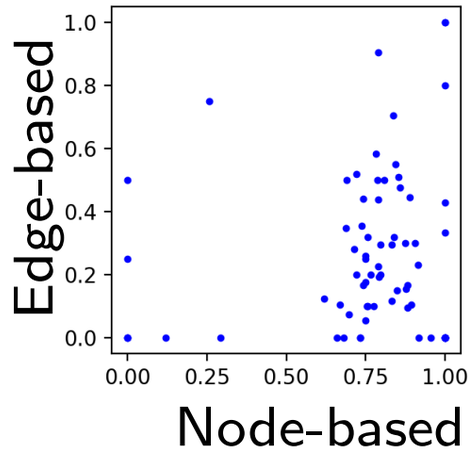
Spring embedder

Problems

- Weak correlation







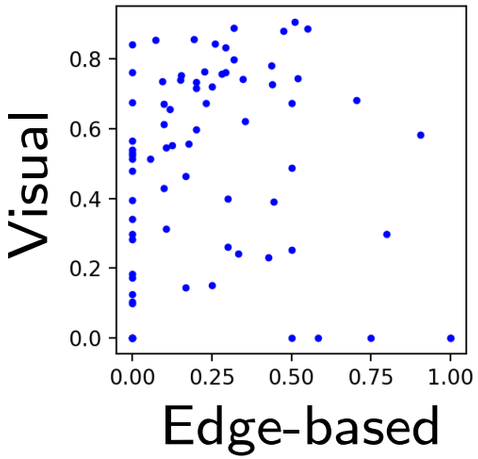
70 random graphs of size 5-12

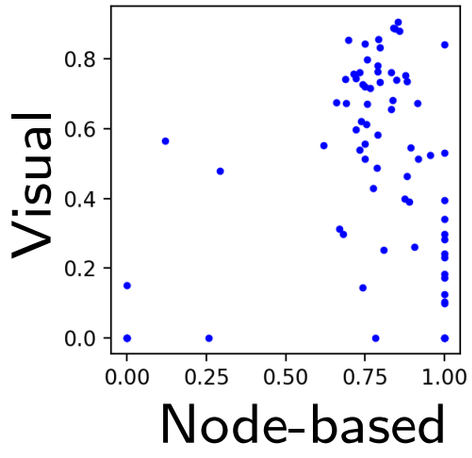
Spring embedder

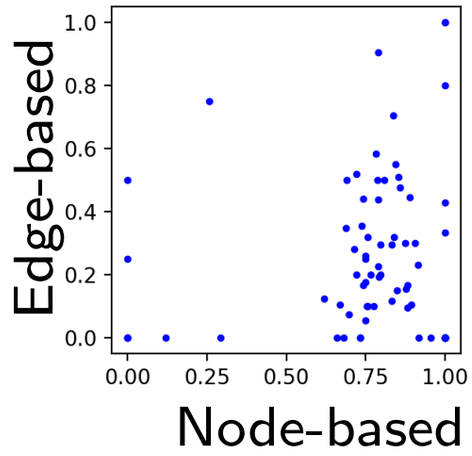
- Hard to optimize parameters

Problems

- Weak correlation







70 random graphs of size 5-12

Spring embedder

- Hard to optimize parameters
- Which metric to use in which case?

Summary

Features

- Loading standard benchmark datasets
- Embedding manipulation
  - Planarization
  - Combinatorial embedding
- Heatmaps



gdMetriX

Implemented metrics

Node distribution
Center of mass
Closest pair of points
Closest pair of elements
Concentration
Homogeneity
Balance
Node orthogonality
Gabriel ratio

Boundary
Area
Aspect ratio

Crossings
Crossings
Crossing angles
Crossing angular resolution

Edge directions
Angular resolution
Average flow
Upwards flow
Edge orthogonality

Symmetry
Node-based
Edge-based
Visual