Karlsruhe Institute of Technology

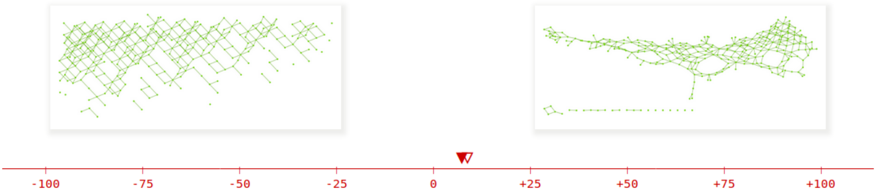
Aesthetic Value of Graph Layouts

Investigation of Statistical Syndromes for Automatic Quantification

Master’s Thesis of Moritz Klammler at the Institute of Theoretical Computer Science

Result of Comparing Layouts c240f3cc and c9cc5217 (p = 7.443)

Compare the feature vectors of the two layouts.



Abstract

Visualizing relational data as drawing of graphs is a technique in very wide-spread use across many fields and professions. While many graph drawing algorithms have been proposed to automatically generate a supposedly high-quality picture from an abstract mathematical data structure, the graph drawing community is still searching for a way to quantify the aesthetic value of any given solution in a way that allows one to compare graph layouts created by different algorithms for the same graph (presumably to automatically choose the better one). We believe that one promising path towards this goal could be enabled by combining data analysis techniques that have proven useful in other scientific disciplines that are dealing with large structures such as astronomy, crystallography or thermodynamics. In this work we present an initial investigation of some statistical properties of graph layouts that we believe could provide viable syndromes for the aesthetic value. As a proof of concept, we used machine learning techniques to train a neural network with the results of our data analysis and thereby built a model that is able to discriminate between better and worse layouts with an accuracy of 95 %. A rudimentary evaluation of the model was performed and is presented. This work primarily provides an infrastructure to enable further experimentation on the topic and will be made available to the public as Free Software.

Contents

Introduction
- Motivation
- Problem Statement
- Previous Work
- Our Contribution

Methodology
- Implementation

Statistical Syndromes

Data Generation
- Graphs
- Layouts

Data Augmentation
- Layout Worsening
- Layout Interpolation

Feature Extraction
- Entropy of Histograms
- Entropy of Sliding Averages

Discriminator Model

Evaluation
- Accuracy
- Contribution of Individual Properties

Conclusions & Future Work
- Summary
- Open Questions and Future Plans

Bibliography

Contents

Introduction
- Motivation
- Problem Statement
- Previous Work
- Our Contribution

Methodology

Statistical Syndromes

Data Generation

Data Augmentation

Feature Extraction

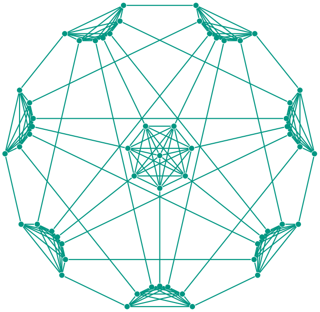
Discriminator Model

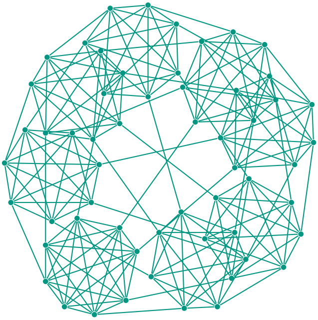
Evaluation

Conclusions & Future Work

Bibliography

Motivation





Possible applications:

- Run N layout algorithms in parallel, choose the best result
- Select layout algorithm for a given application
- Aid the development of domain-specific layout algorithms

Problem Statement

Karlsruhe Institute of Technology

- *Simple graphs* only[1]
- *Vertex layouts* only[2]
- Two layouts for the *same* graph given
- Decide which one is aesthetically more pleasing[3]

[1] undirected, no loops, no multiple edges
[2] vertices are 2D points, edges are straight lines
[3] ideally, we aim for a *partial order*

Previous Work

Existing Measures:

- Energy (from force-directed methods)
- Purchase (2002)

  edge crossings, edge bends, symmetry, minimal angles between edges, edge orthogonality, node orthogonality, consistent flow direction
- Binary Stress (Kamada and Kawai 1989; Koren and Çivril 2008)
- Klapaukh (2014)
- Combined Metric (Huang et al. 2016)

Problems with these:

- Too many a priory assumptions
- Too localized / too little context
- Might loose valuable information due to oversimplification
- Some are unstable with respect to the simplest transformations (scaling)

Our Contribution

- Idea: Use data analysis techniques that were successful in other disciplines (crystallography, astronomy, thermodynamics, ...)
- Strategy: Condense this information into a fixed-size feature vector via statistic analysis
- Question: Can we find syndromes that allow for successful automatic quantification?
- Guideline: Try to use as few a priori assumptions as possible and detect features from first principles

Neural Network Demo - Mozilla Firefox

Neural Network Demo

localhost:8000/nn/demo/?lhs=0f76aebc&

Neural Network Demo

Compare *N* random pairs of layouts:

number [Compare]

Compare two specific layouts (IDs may be abbreviated to unambiguous prefixes of four or more digits):

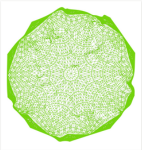
0f76aebcfafc504fa057b9d1f39955fb | 33eacef0264f234f10835b1427e58383 | [Compare]

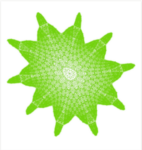
Performance

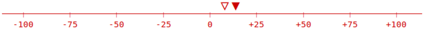
Loading Model	0.546
Evaluating Model	0.044
Total Time	0.590

Result of Comparing Layouts 0f76aebc and 33eacef0 (*p* = 13.997)

Compare the feature vectors of the two layouts.







Contents

Introduction

Methodology
- Implementation

Statistical Syndromes

Data Generation

Data Augmentation

Feature Extraction

Discriminator Model

Evaluation

Conclusions & Future Work

Bibliography

Methodology

Karlsruhe Institute of Technology

- Gather as many graphs as possible
- Obtain as many (known) good and bad layouts as possible
- Compute properties for all of them
- Use *a priori* knowledge to label pairs of layouts
- Use data to build a discriminator

Implementation

- Plug-in infrastructure for unattended experimentation
- Graph generators, layouts, layout transformations and properties implemented as small programs
- Siamese neural network
- Feature-rich web front-end for data inspection

Technologies:

- Open Graph Drawing Framework (OGDF)
- Keras + TensorFlow
- C++, Python, SQLite, XSLT, CMake, …

Contents

Introduction

Methodology

Statistical Syndromes

Data Generation

Data Augmentation

Feature Extraction

Discriminator Model

Evaluation

Conclusions & Future Work

Bibliography

Statistical Syndromes of Graph Layouts

Karlsruhe Institute of Technology

We investigated several *properties* (multisets of scalar events) for a given layout.

- Principal Components (PRINCOMP1ST and PRINCOMP2ND)
- Angles Between Incident Edges (ANGULAR)
- Edge Lengths (EDGE_LENGTH)
- Pairwise Distances (RDF_GLOBAL and RDF_LOCAL)
- Tension (TENSION)

Principal Components

Karlsruhe Institute of Technology

- Find linear independent axes among which the moment of inertia is maximized
- Consider the distribution of vertex coordinates along those axes
- Can be computed with $\mathcal{O}(n)$ effort

$$\mathrm{PRINCOMP1ST} = \left[\left\langle p^{(1)} \middle| \Gamma(v) \right\rangle : v \in V \right]$$
$$\mathrm{PRINCOMP2ND} = \left[\left\langle p^{(2)} \middle| \Gamma(v) \right\rangle : v \in V \right]$$

Principal Components

- Find linear independent axes among which the moment of inertia is maximized
- Consider the distribution of vertex coordinates along those axes
- Can be computed with $\mathcal{O}(n)$ effort

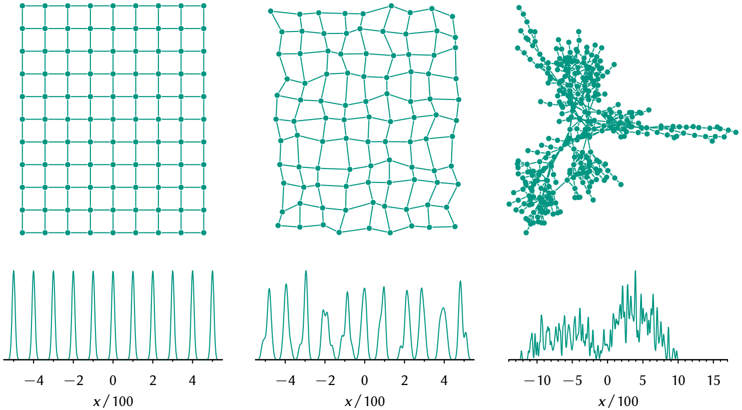
first principal axis

$$\mathrm{PRINCOMP1ST} = \left[\left\langle p^{(1)} \middle| \Gamma(v) \right\rangle : v \in V\right]$$

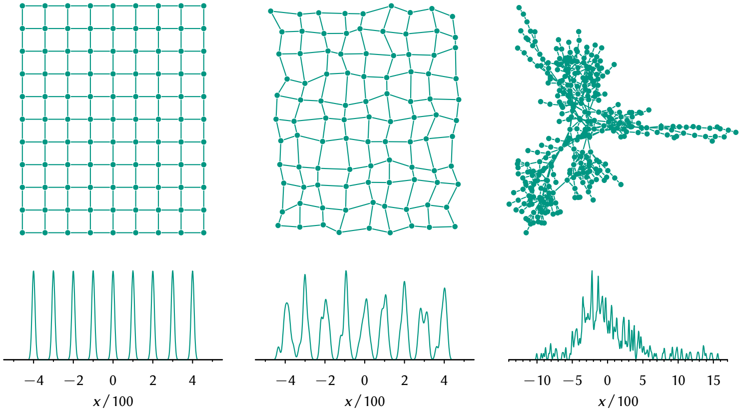
$$\mathrm{PRINCOMP2ND} = \left[\left\langle p^{(2)} \middle| \Gamma(v) \right\rangle : v \in V\right]$$

second principal axis

1st Principal Component (PRINCOMP1ST)



2^{nd} Principal Component (PRINCOMP2ND)

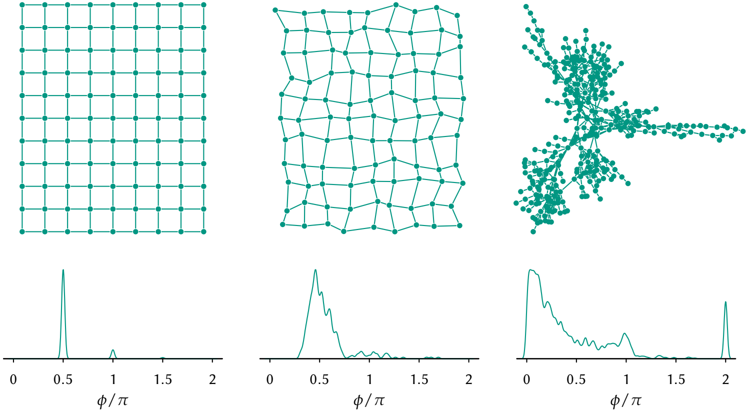


Angles Between Incident Edges (ANGULAR)

- Enumerate polar angles of incident edges in clockwise order
- Compute adjacent differences (of polar angles)
- Special cases for $\deg(v) = 1$ and degenerate cases
- Consider distribution of all those angles
- Can be computed with $\mathcal{O}(n + m)$ effort

$$\text{ANGULAR} = \bigcup_{v \in V} \phi_\Gamma(v)$$

Angles Between Incident Edges (ANGULAR)



Edge Lengths (EDGE_LENGTH)

- Consider distribution of edge lengths
- Can be computed with $\mathcal{O}(m)$ effort

$$\mathrm{EDGE\_LENGTH} = \left[\mathsf{length}_\Gamma(e) : e \in E\right]$$

Edge Lengths (EDGE_LENGTH)

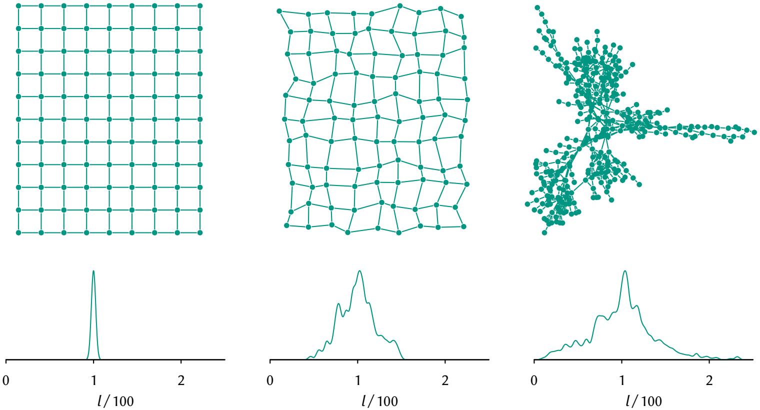
Karlsruhe Institute of Technology

- Consider distribution of edge lengths
- Can be computed with $\mathcal{O}(m)$ effort

$$\text{EDGE\_LENGTH} = \left[\text{length}_{\Gamma}(e) : e \in E\right]$$

length of edge e in layout Γ

Edge Lengths (EDGE_LENGTH)



Pairwise Distances (RDF_GLOBAL)

- Compute pairwise distances between all pairs of vertices
- Can be computed with $\mathcal{O}(n^2)$ effort

$$\mathrm{RDF\_GLOBAL} = \left[\mathsf{dist}_{\Gamma}(v_1, v_2) : v_1, v_2 \in V\right]$$

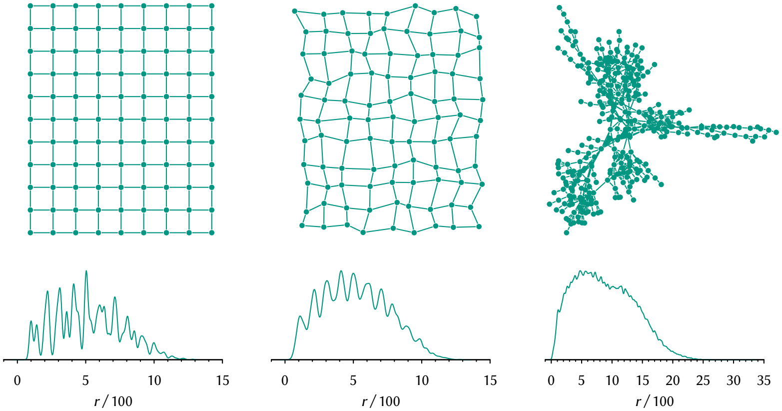
Pairwise Distances (RDF_GLOBAL)

- Compute pairwise distances between all pairs of vertices
- Can be computed with $\mathcal{O}(n^2)$ effort

$$\text{RDF\_GLOBAL} = \left[\text{dist}_\Gamma(v_1, v_2) : v_1, v_2 \in V\right]$$

distance between $\Gamma(v_1)$ and $\Gamma(v_2)$

Pairwise Distances (RDF_GLOBAL)



Pairwise Distances (RDF_LOCAL(d))

- Restrict global RDF to those pairs of vertices that have a graph-theoretical distance of at most $d \in \mathbb{R}$
- Repeat for different values of $d = 2^i$ for $i \in \mathbb{N}_0$ up to the longest finite shortest path in the graph
- Interpolates between EDGE_LENGTH and RDF_GLOBAL
- Can be computed with $\mathcal{O}(n^3)$ effort

$$\text{RDF\_LOCAL}(d) = \left[\mathrm{dist}_\Gamma(v_1, v_2) : \mathrm{dist}(v_1, v_2) \leq d : v_1, v_2 \in V\right]$$

Pairwise Distances (RDF_LOCAL(d))

Karlsruhe Institute of Technology

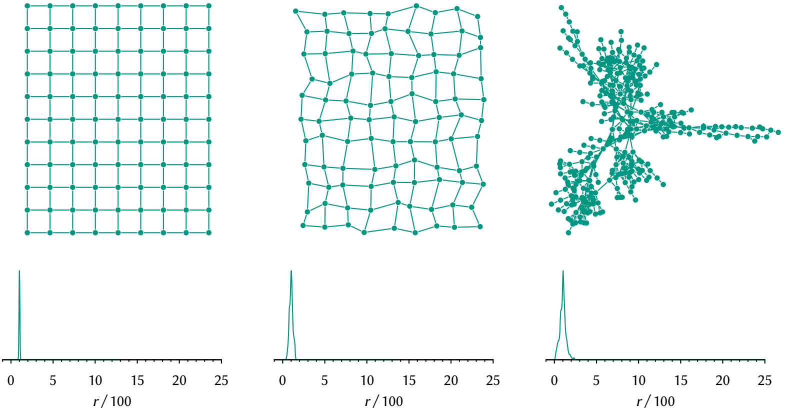
- Restrict global RDF to those pairs of vertices that have a graph-theoretical distance of at most $d \in \mathbb{R}$
- Repeat for different values of $d = 2^i$ for $i \in \mathbb{N}_0$ up to the longest finite shortest path in the graph
- Interpolates between EDGE_LENGTH and RDF_GLOBAL
- Can be computed with $\mathcal{O}(n^3)$ effort

$$\text{RDF\_LOCAL}(d) = [\mathsf{dist}_\Gamma(v_1, v_2) : \mathsf{dist}(v_1, v_2) \leq d : v_1, v_2 \in V]$$

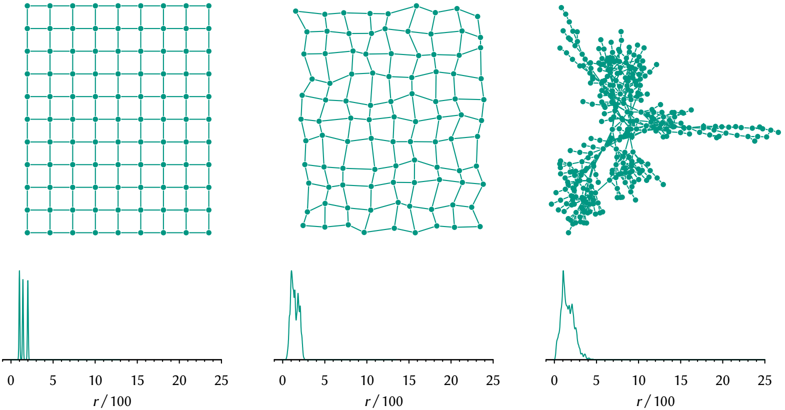
distance between $\Gamma(v_1)$ and $\Gamma(v_2)$ → $\mathsf{dist}_\Gamma(v_1, v_2)$

length of shortest path from v_1 to v_2 → $\mathsf{dist}(v_1, v_2)$

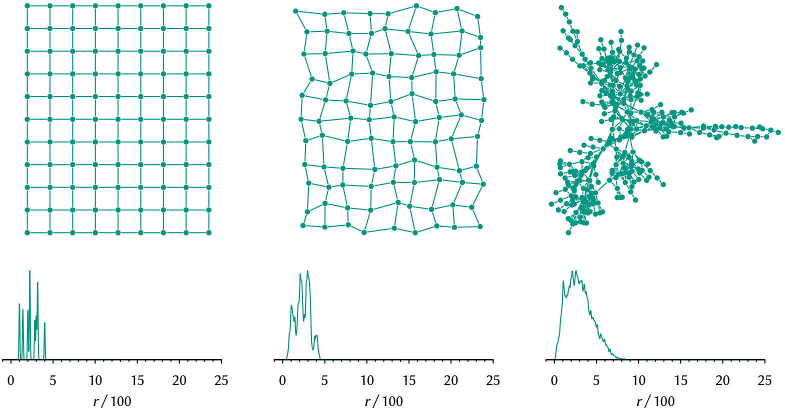
Pairwise Distances (RDF_LOCAL(1))



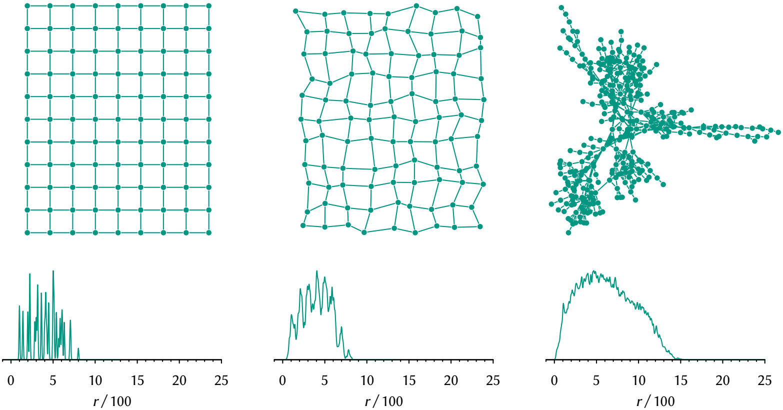
Pairwise Distances (RDF_LOCAL(2))



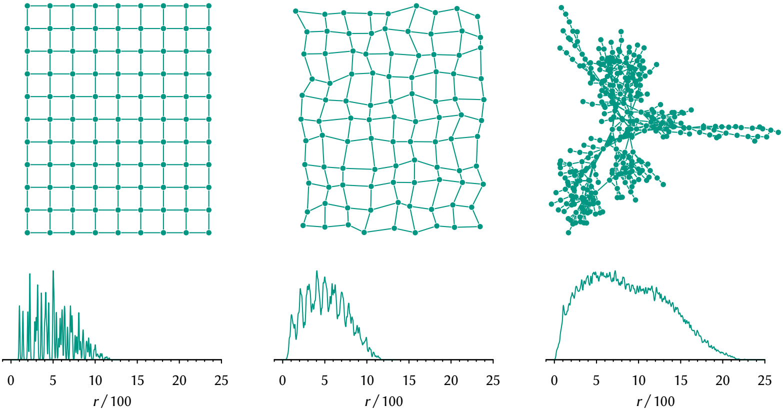
Pairwise Distances (RDF_LOCAL(4))



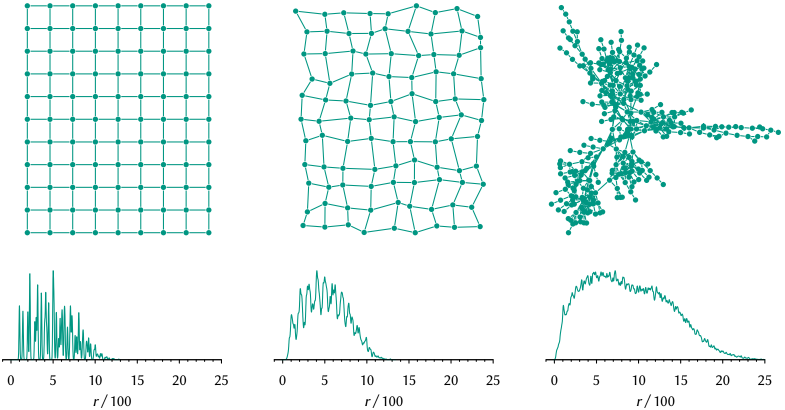
Pairwise Distances (RDF_LOCAL(8))



Pairwise Distances (RDF_LOCAL(16))



Pairwise Distances (RDF_LOCAL(32))



Tension (TENSION)

- Consider distribution of quotients of node and graph distances
- Inspired by stress but has well-behaved response to scaling
- Can be computed with $\mathcal{O}(n^3)$ effort

$$\mathtt{TENSION} = \frac{|E|}{\sum_{e \in E} \mathrm{length}_\Gamma(e)} \left[\frac{\mathrm{dist}_\Gamma(v_1, v_2)}{\mathrm{dist}(v_1, v_2)} : v_1, v_2 \in V \right]$$

Tension ($\mathtt{TENSION}$)

Karlsruhe Institute of Technology

- Consider distribution of quotients of node and graph distances
- Inspired by stress but has well-behaved response to scaling
- Can be computed with $\mathcal{O}(n^3)$ effort

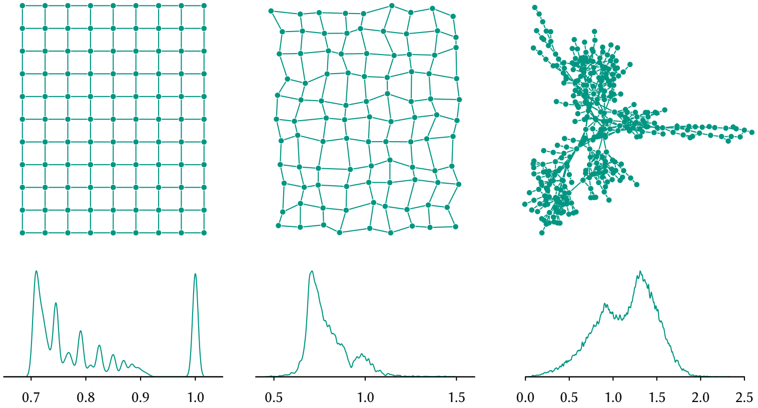
distance between $\Gamma(v_1)$ and $\Gamma(v_2)$

$$\mathtt{TENSION} = \frac{|E|}{\sum_{e \in E} \mathrm{length}_\Gamma(e)} \left[\frac{\mathrm{dist}_\Gamma(v_1, v_2)}{\mathrm{dist}(v_1, v_2)} : v_1, v_2 \in V \right]$$

length of edge e in layout Γ

length of shortest path from v_1 to v_2

Tension (TENSION)



Contents

Introduction

Methodology

Statistical Syndromes

Data Generation
- Graphs
- Layouts

Data Augmentation

Feature Extraction

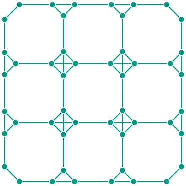
Discriminator Model

Evaluation

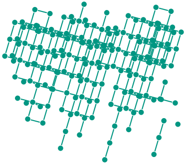
Conclusions & Future Work

Bibliography

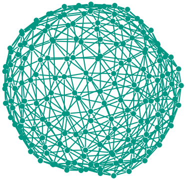
Graph Generators



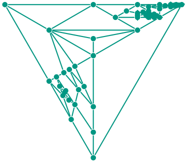
LINDENMAYER



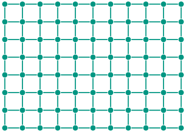
QUASI4D



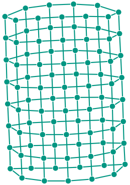
BOTTLE



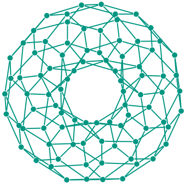
MOSAIC1



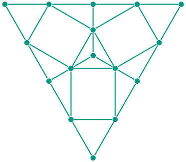
GRID



TORUS1

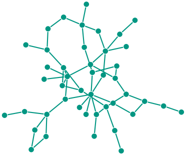


TORUS2

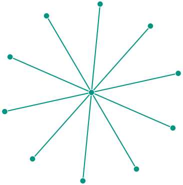


MOSAIC2

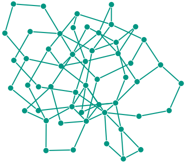
Graph Import Sources



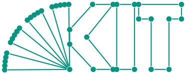
ROME



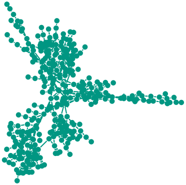
NORTH



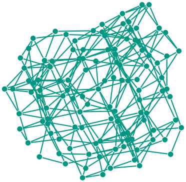
RANDDAG



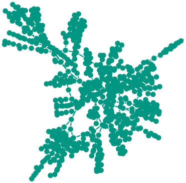
IMPORT



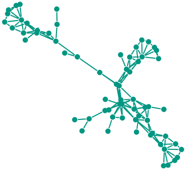
BCSPWR



GRENOBLE

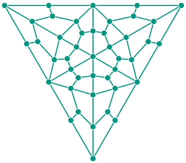


PSADMIT

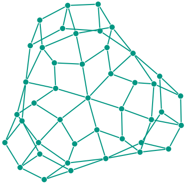


SMTAPE

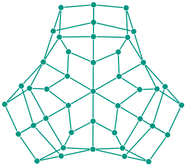
Layouts



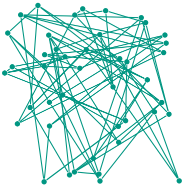
NATIVE



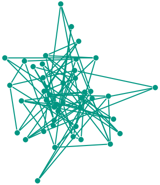
FMMM



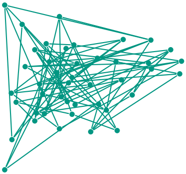
STRESS



RANDOM_UNIFORM



RANDOM_NORMAL



PHANTOM

Contents

Introduction

Methodology

Statistical Syndromes

Data Generation

Data Augmentation

Layout Worsening

Layout Interpolation

Feature Extraction

Discriminator Model

Evaluation

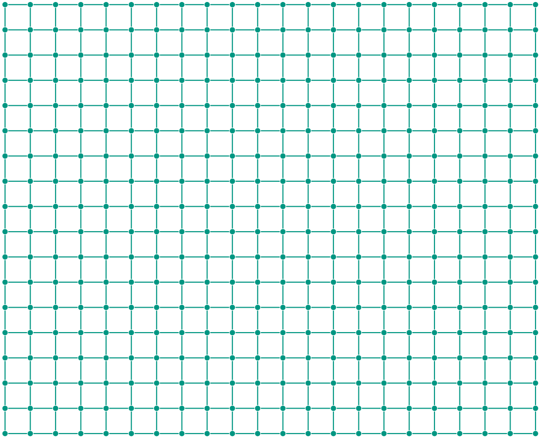
Conclusions & Future Work

Bibliography

Data Augmentation

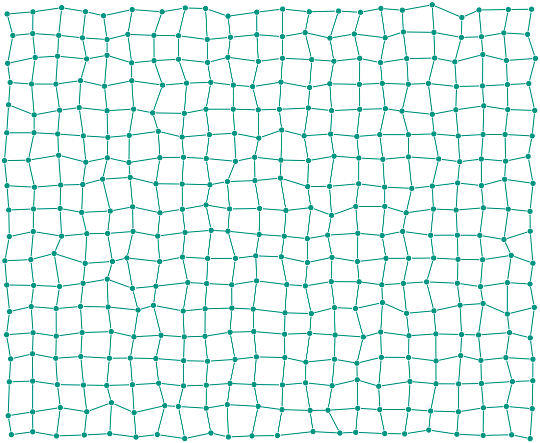
- Layout worsening (unary layout transformation)
  - Input: Parent layout Γ and parameter $0 \leq r \leq 1$
  - Output: Worsened layout Γ'_r
- Layout interpolation (binary layout transformation)
  - Input: Parent layouts Γ_A and Γ_B and parameter $0 \leq r \leq 1$
  - Output: Interpolated layout Γ'_r

Layout Worsening (PERTURB)

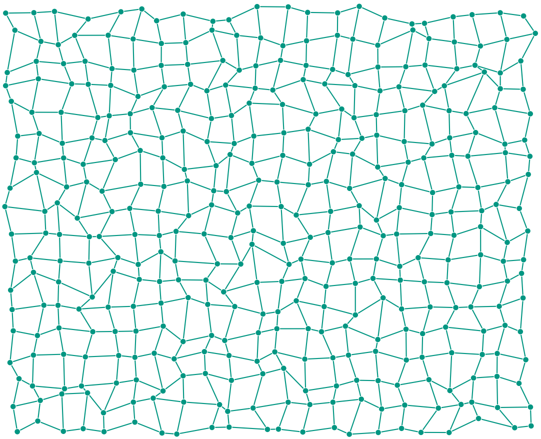


Layout Worsening (PERTURB)

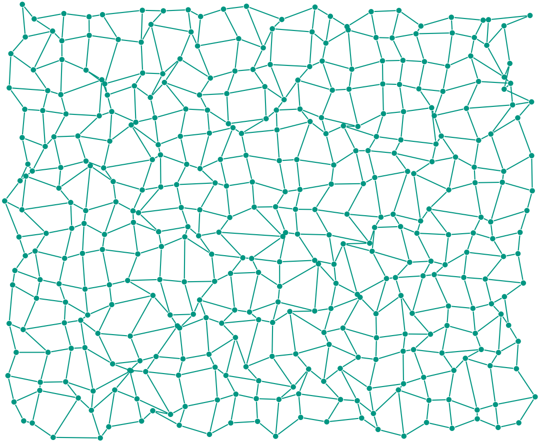
Karlsruhe Institute of Technology



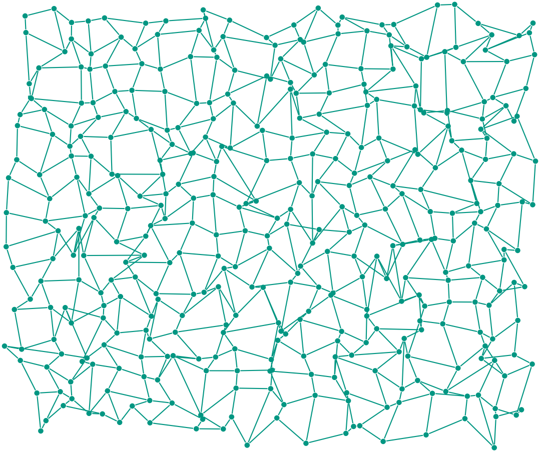
Layout Worsening (PERTURB)



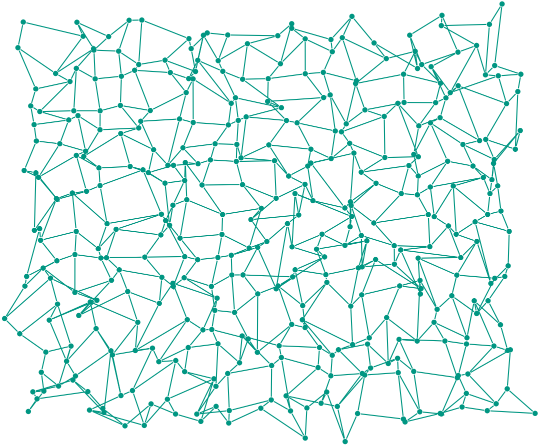
Layout Worsening (PERTURB)



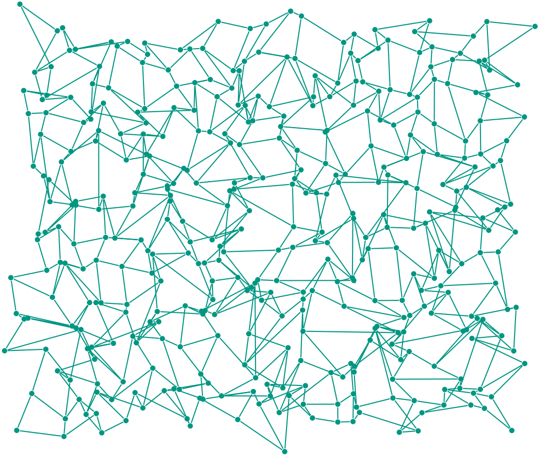
Layout Worsening (PERTURB)



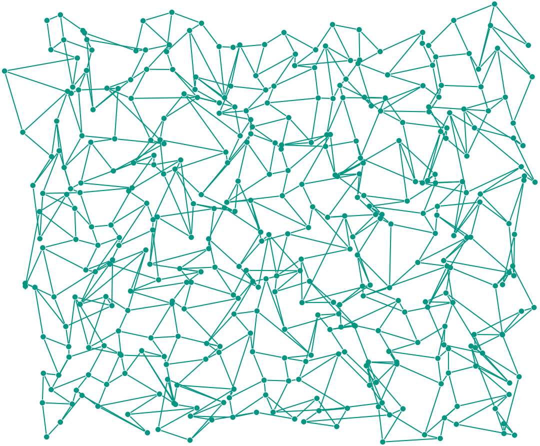
Layout Worsening (PERTURB)



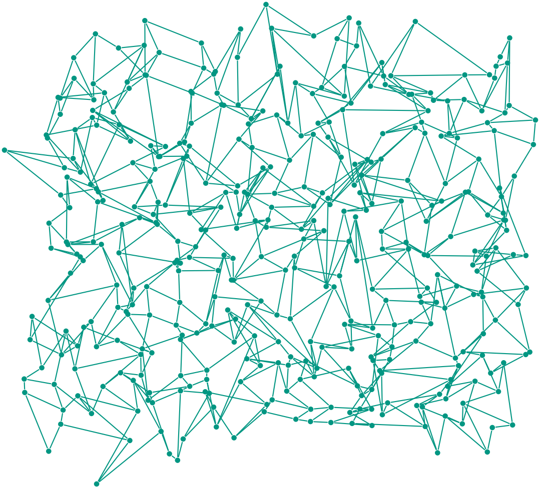
Layout Worsening (PERTURB)



Layout Worsening (PERTURB)

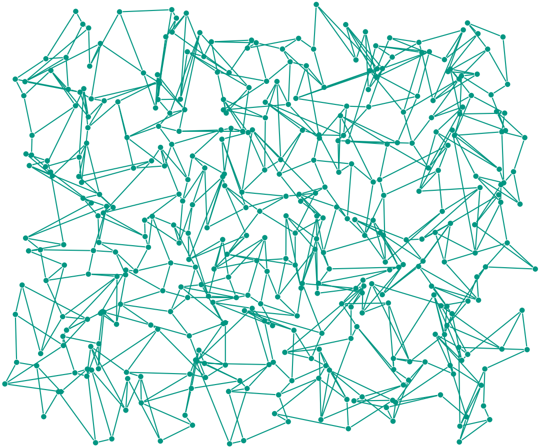


Layout Worsening (PERTURB)



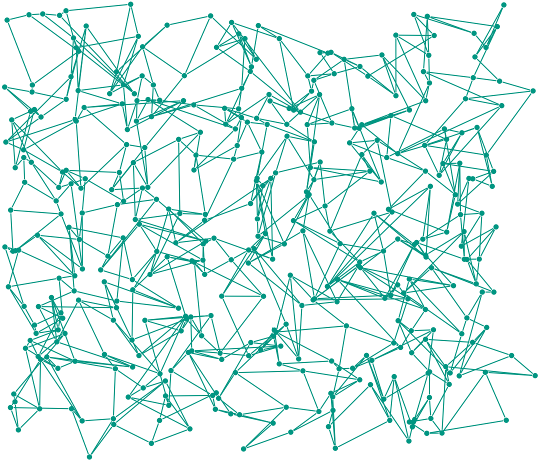
Layout Worsening (PERTURB)

Karlsruhe Institute of Technology

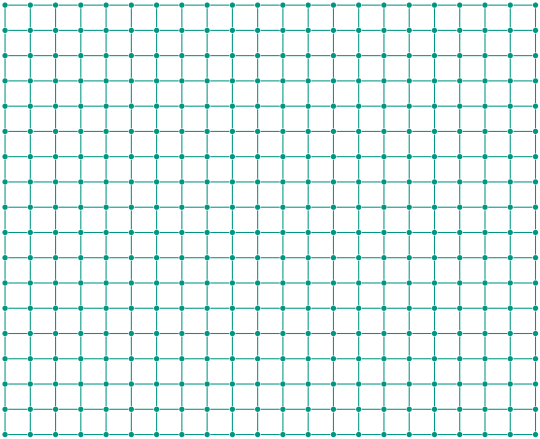


Layout Worsening (PERTURB)

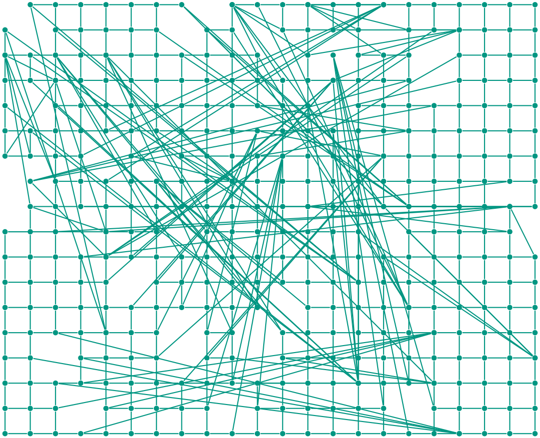
Karlsruhe Institute of Technology



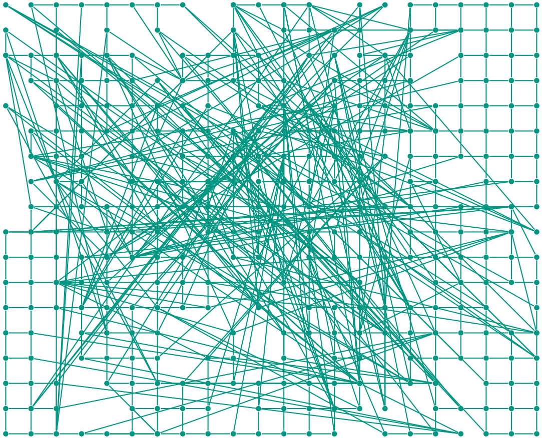
Layout Worsening (`FLIP_NODES`)



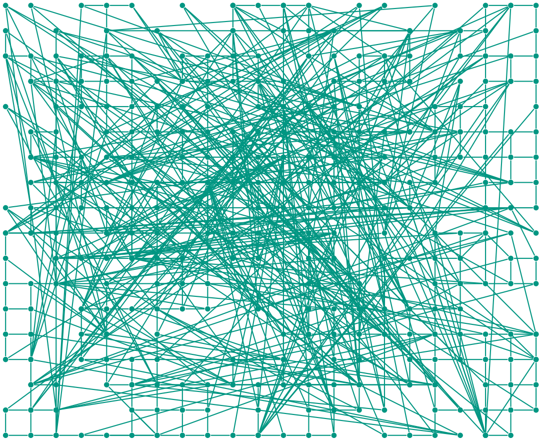
Layout Worsening (FLIP_NODES)



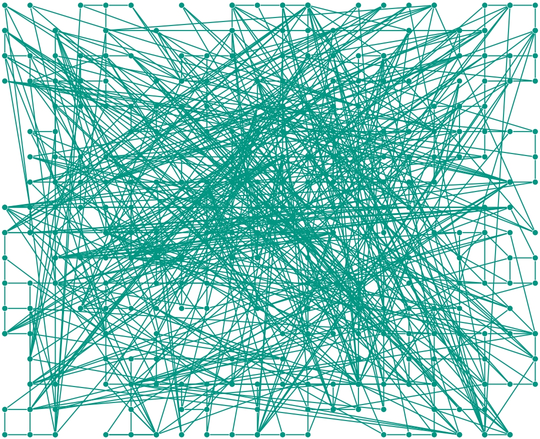
Layout Worsening (FLIP_NODES)



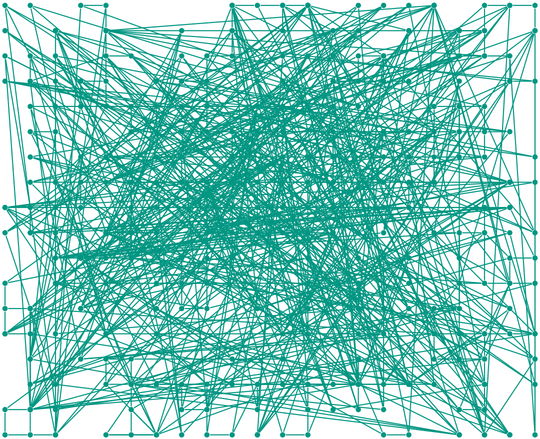
Layout Worsening (FLIP_NODES)



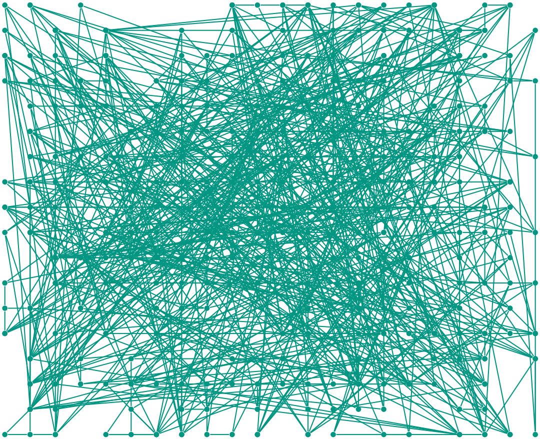
Layout Worsening (`FLIP_NODES`)



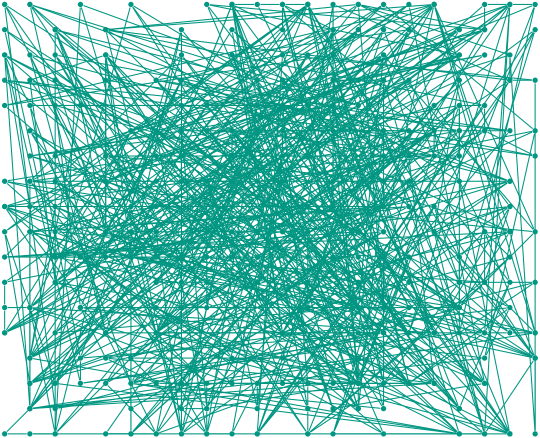
Layout Worsening (FLIP_NODES)



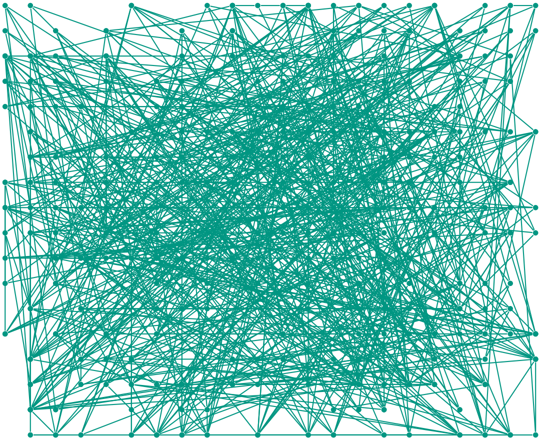
Layout Worsening (FLIP_NODES)



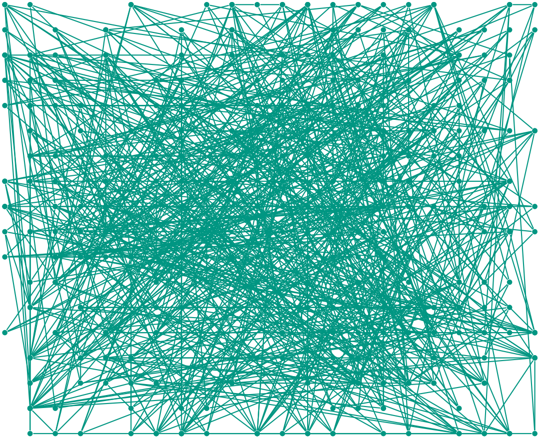
Layout Worsening (FLIP_NODES)



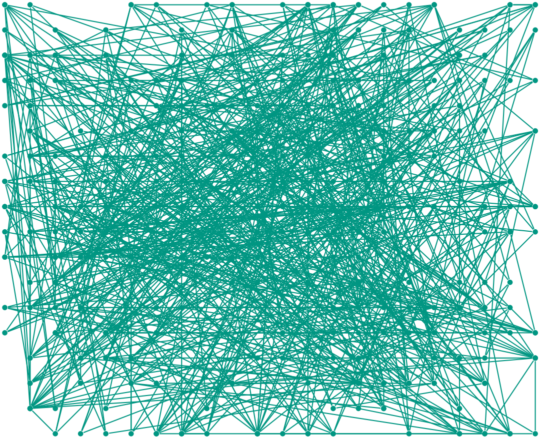
Layout Worsening (FLIP_NODES)



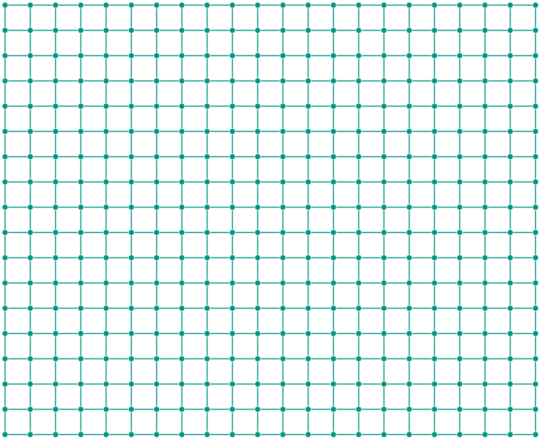
Layout Worsening (FLIP_NODES)



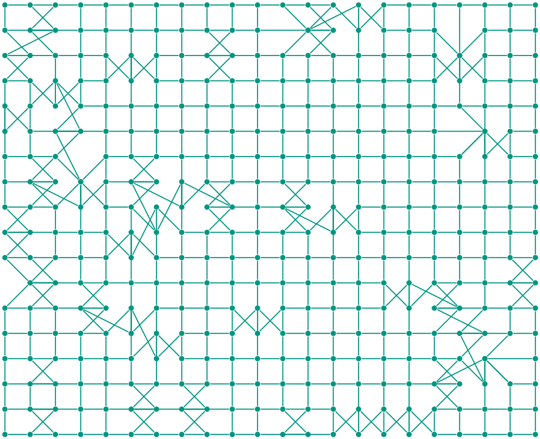
Layout Worsening (FLIP_NODES)



Layout Worsening (FLIP_EDGES)

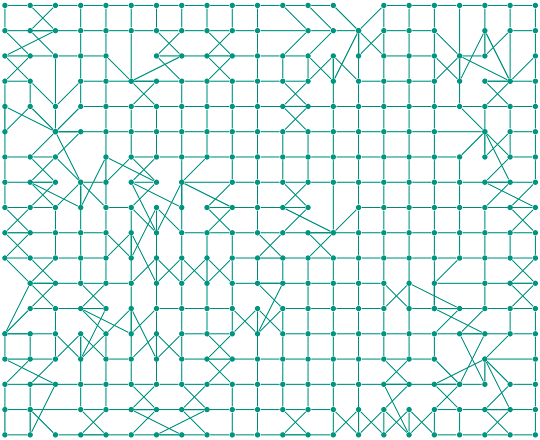


Layout Worsening (`FLIP_EDGES`)

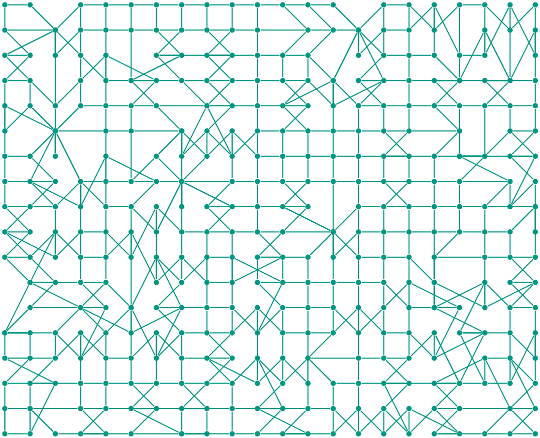


Layout Worsening (FLIP_EDGES)

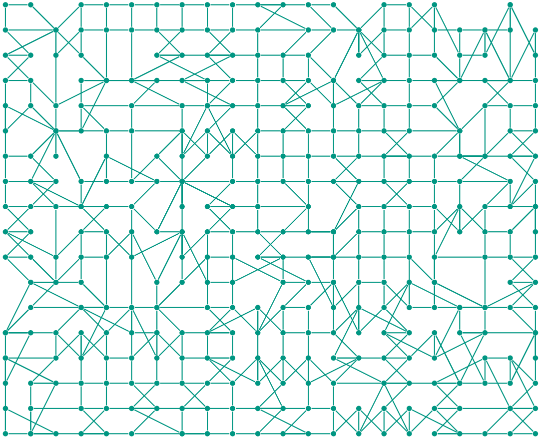
Karlsruhe Institute of Technology



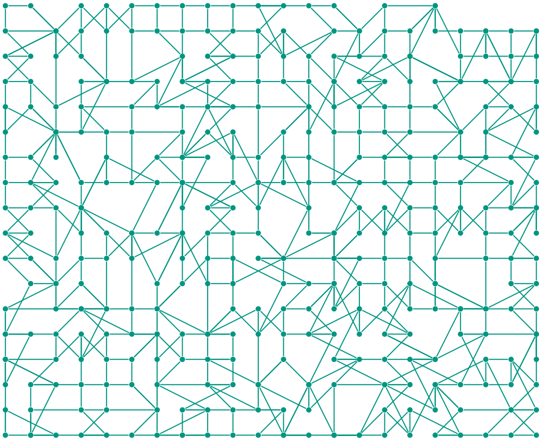
Layout Worsening (FLIP_EDGES)



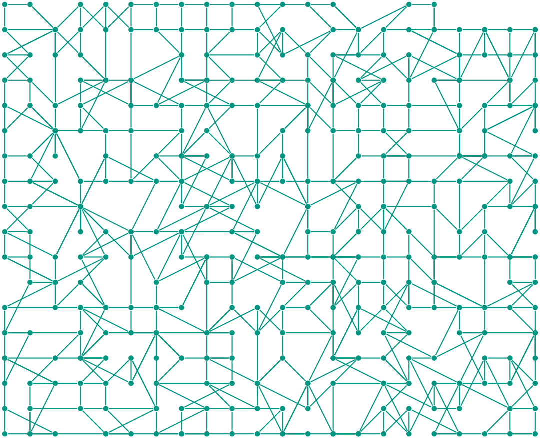
Layout Worsening (FLIP_EDGES)



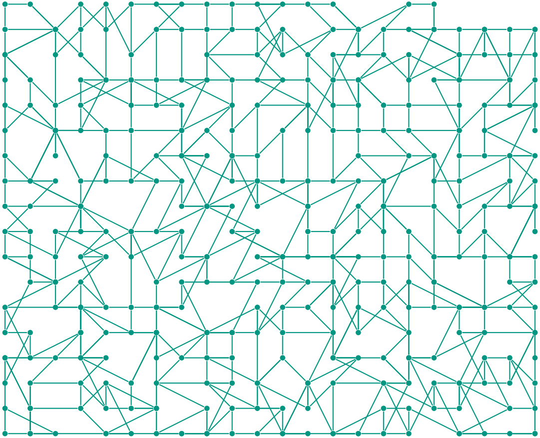
Layout Worsening (FLIP_EDGES)



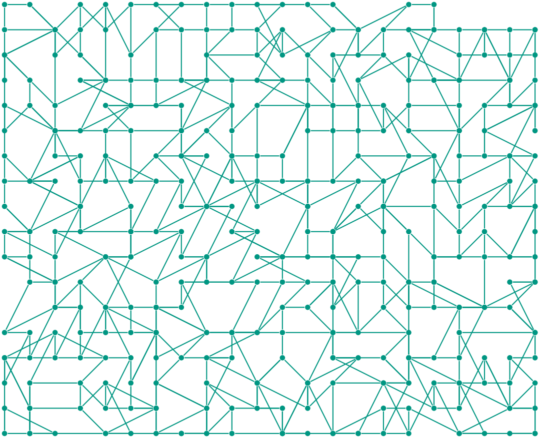
Layout Worsening (FLIP_EDGES)



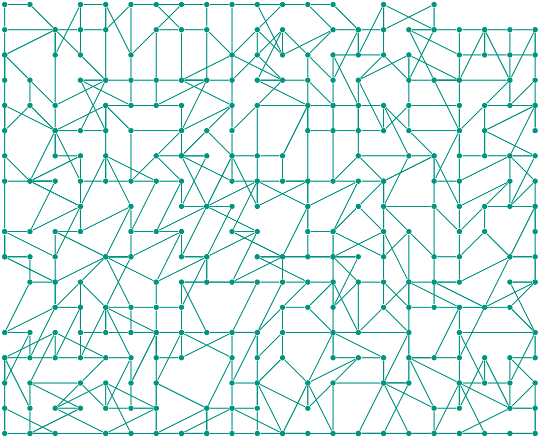
Layout Worsening (FLIP_EDGES)



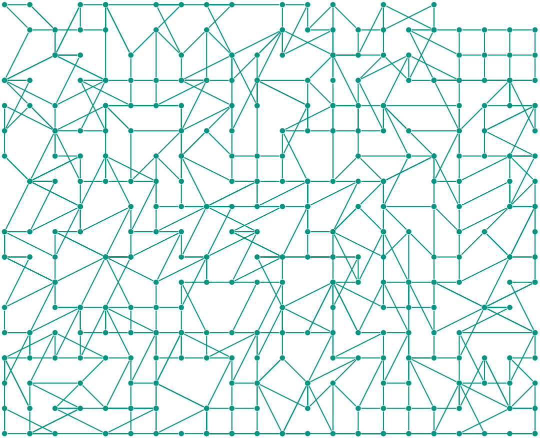
Layout Worsening (`FLIP_EDGES`)



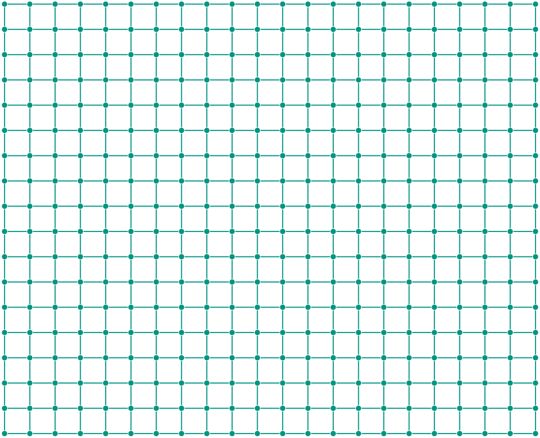
Layout Worsening (FLIP_EDGES)



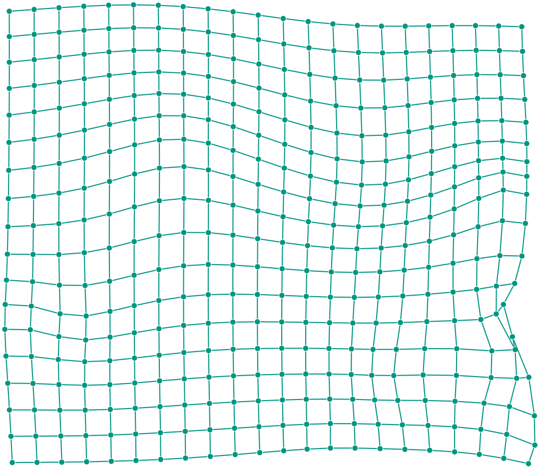
Layout Worsening (FLIP_EDGES)



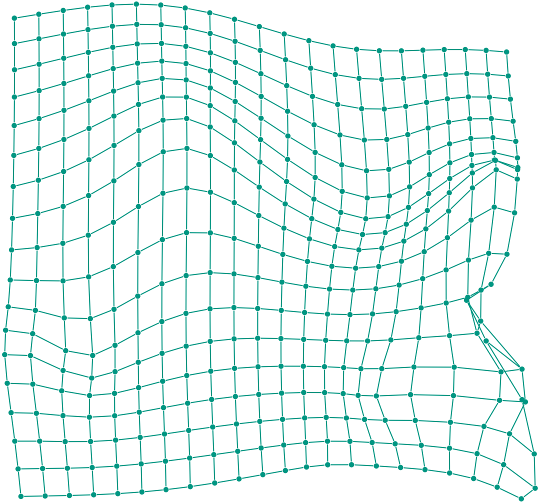
Layout Worsening (MOVLSQ)



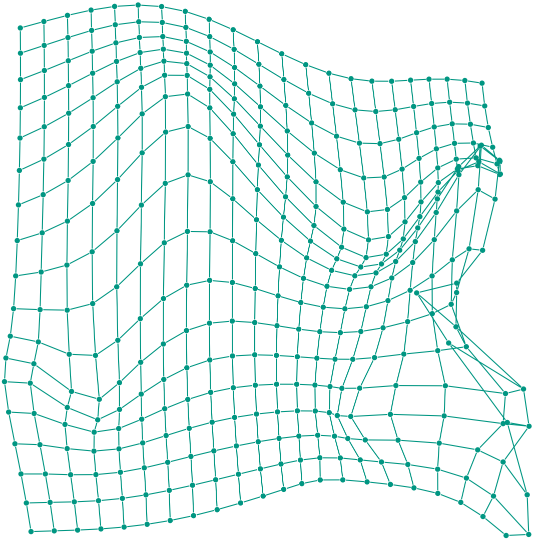
Layout Worsening (MOVLSQ)



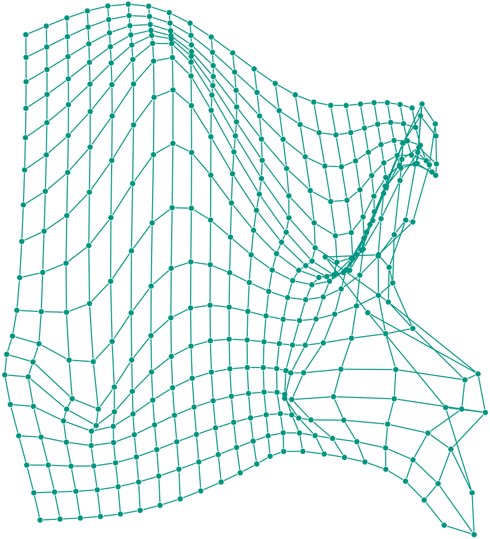
Layout Worsening (MOVLSQ)



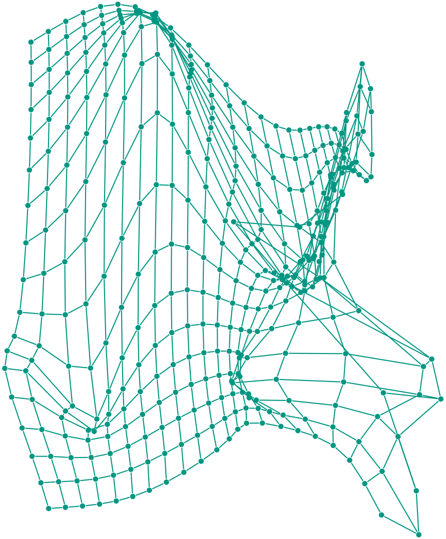
Layout Worsening (MOVLSQ)



Layout Worsening (MOVLSQ)

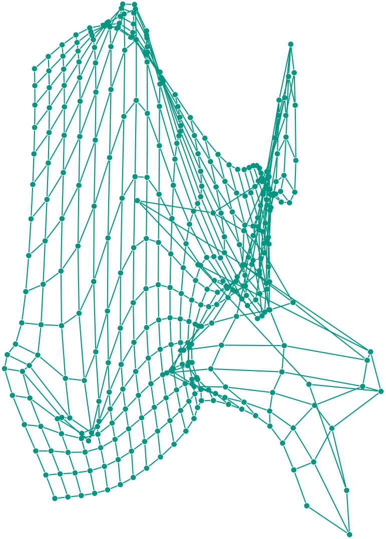


Layout Worsening (MOVLSQ)



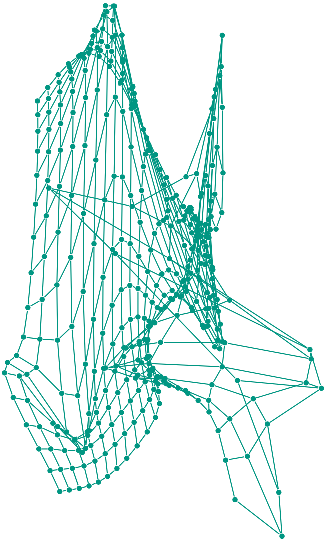
Layout Worsening (MOVLSQ)

Karlsruhe Institute of Technology

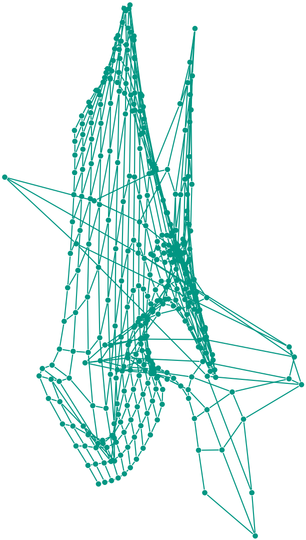


Layout Worsening (MOVLSQ)

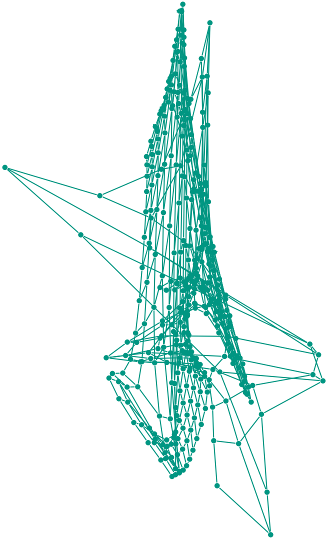
Karlsruhe Institute of Technology



Layout Worsening (MOVLSQ)

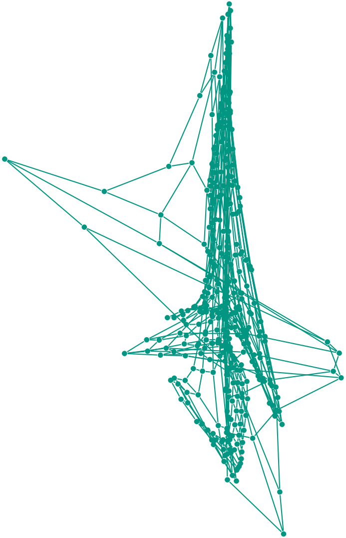


Layout Worsening (MOVLSQ)

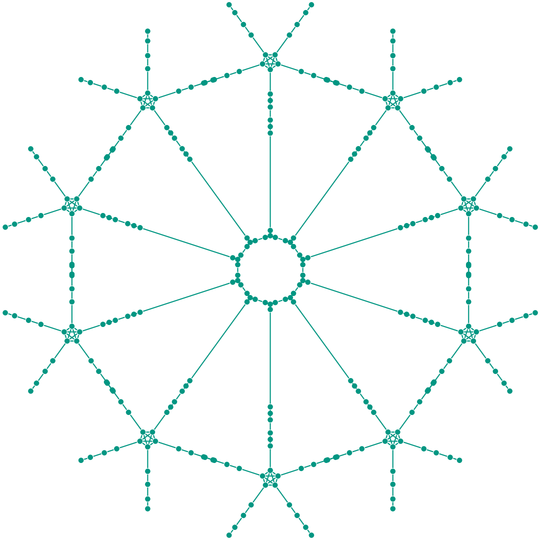


Layout Worsening (MOVLSQ)

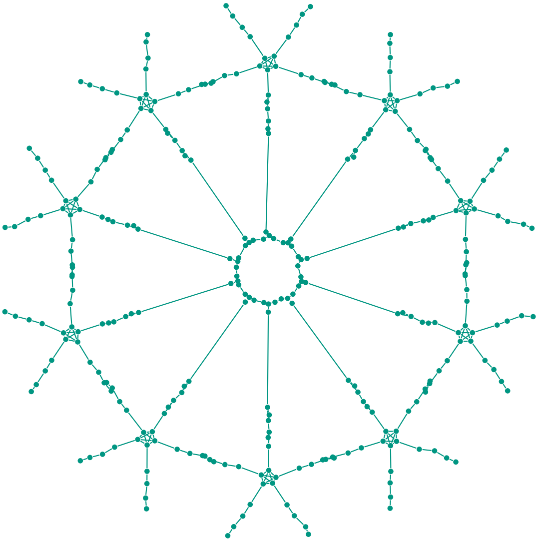
Karlsruhe Institute of Technology



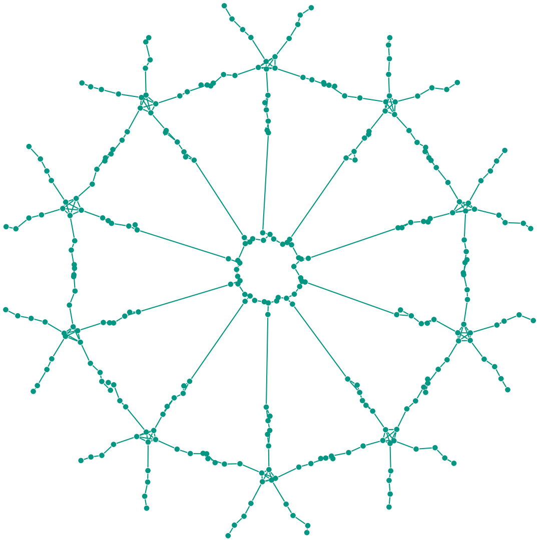
Layout Interpolation (LINEAR)



Layout Interpolation (LINEAR)

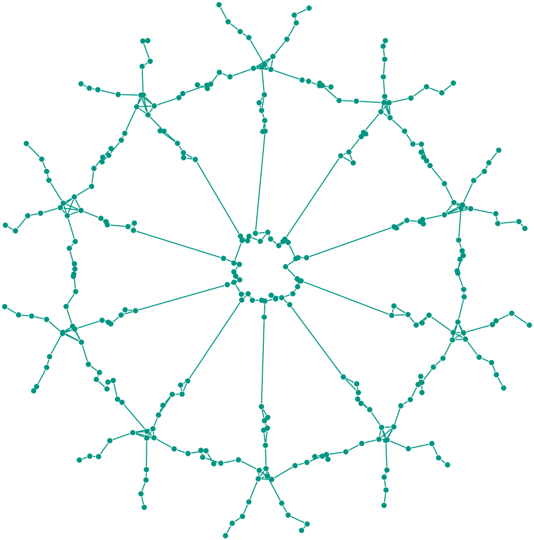


Layout Interpolation (LINEAR)

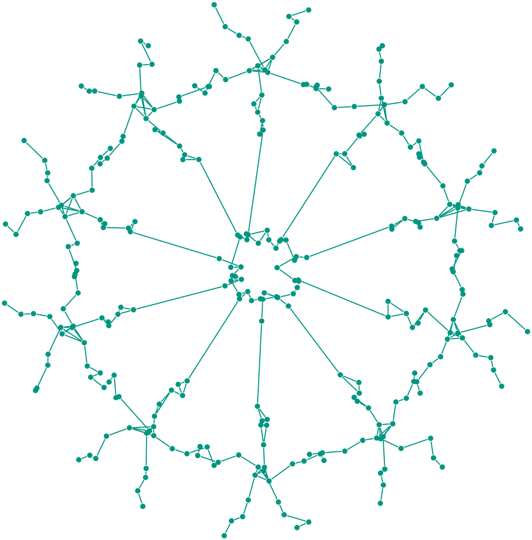


Layout Interpolation (LINEAR)

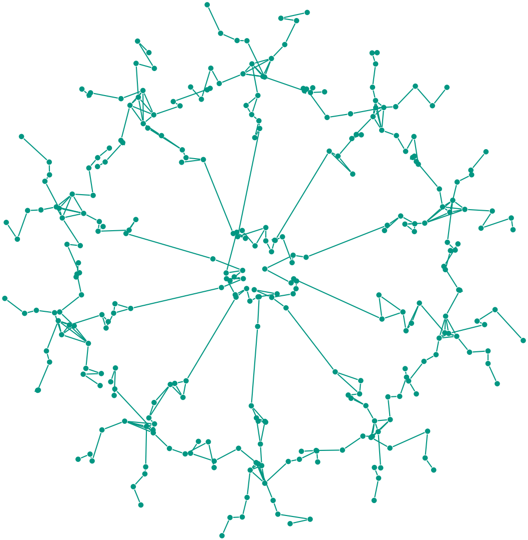
Karlsruhe Institute of Technology



Layout Interpolation (LINEAR)

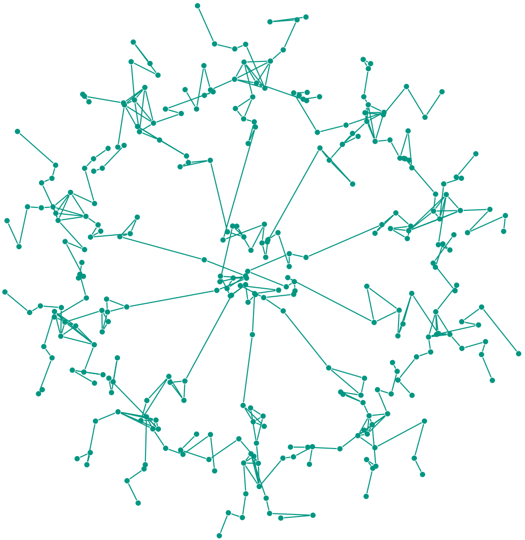


Layout Interpolation (LINEAR)



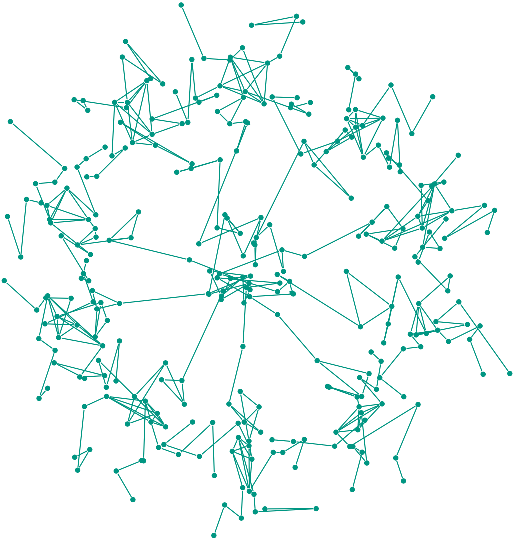
Layout Interpolation (LINEAR)

Karlsruhe Institute of Technology

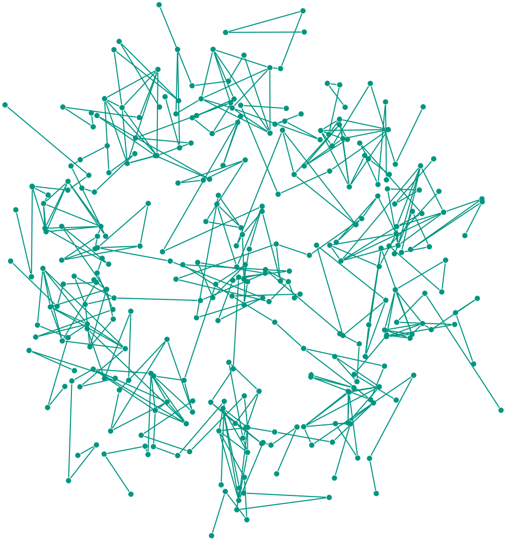


Layout Interpolation (LINEAR)

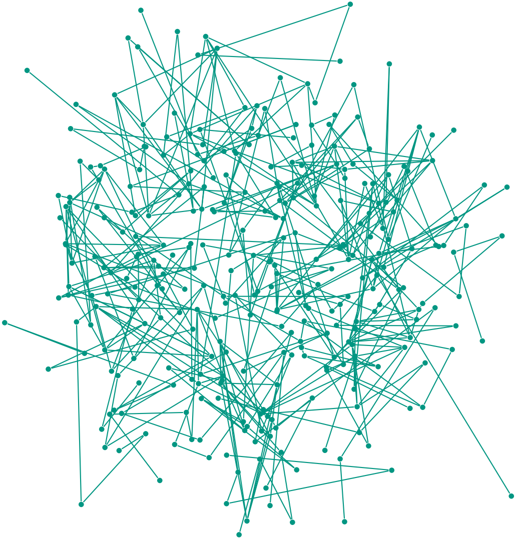
Karlsruhe Institute of Technology



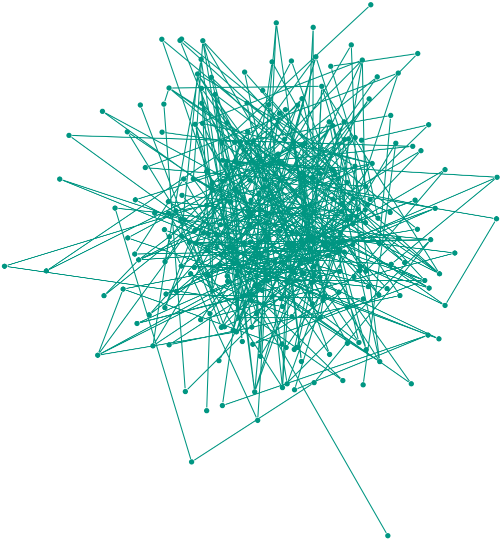
Layout Interpolation (LINEAR)



Layout Interpolation (LINEAR)

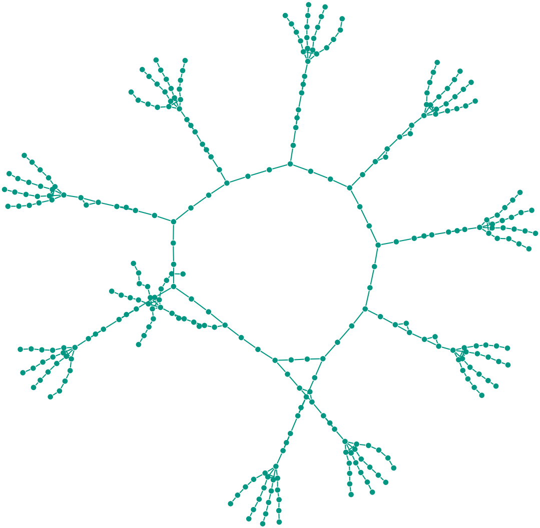


Layout Interpolation (LINEAR)

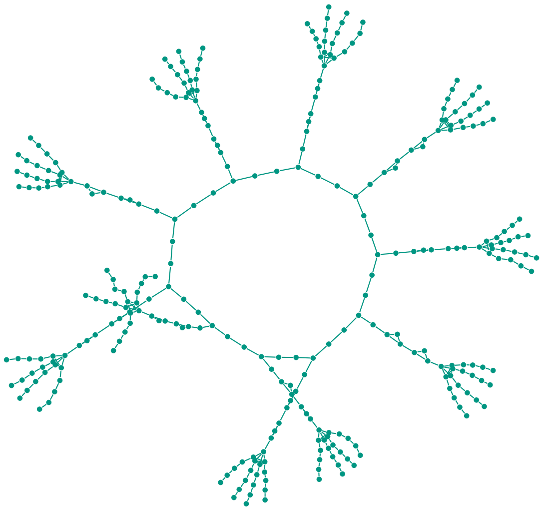


Layout Interpolation (XLINEAR)

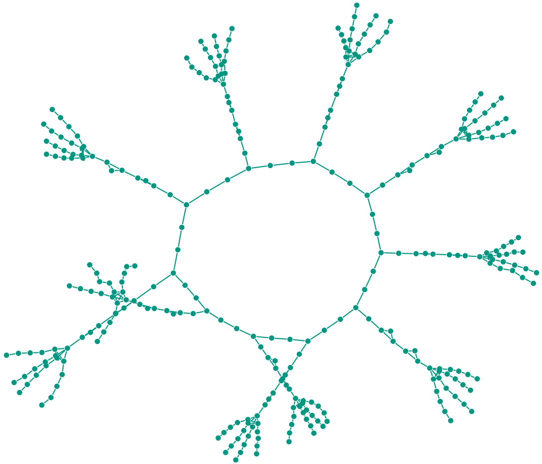
Karlsruhe Institute of Technology



Layout Interpolation (XLINEAR)

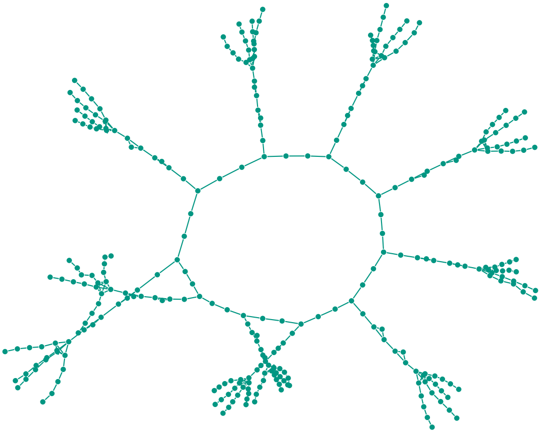


Layout Interpolation (XLINEAR)

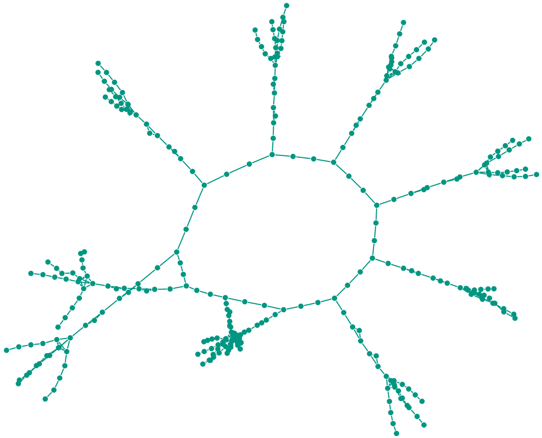


Layout Interpolation (XLINEAR)

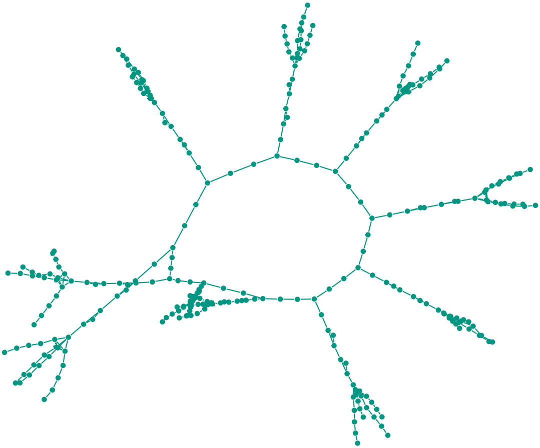
Karlsruhe Institute of Technology



Layout Interpolation (XLINEAR)

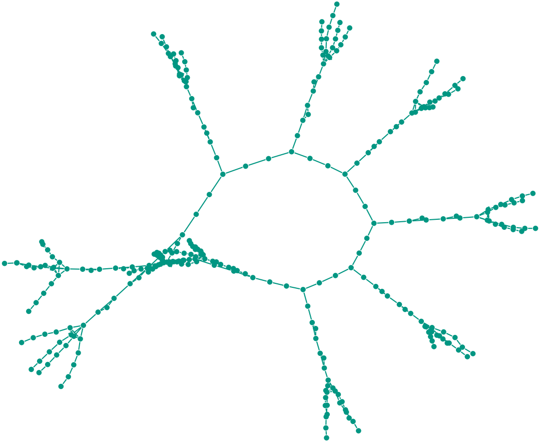


Layout Interpolation (XLINEAR)

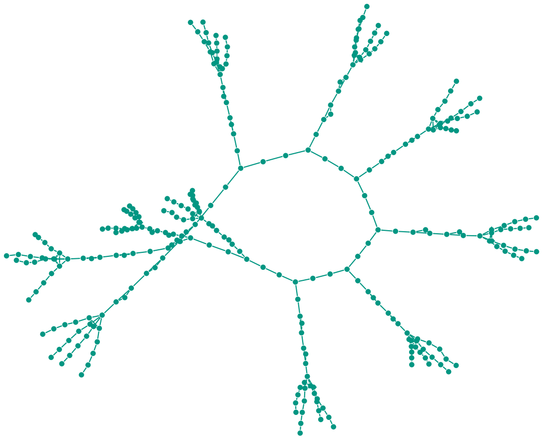


Layout Interpolation (XLINEAR)

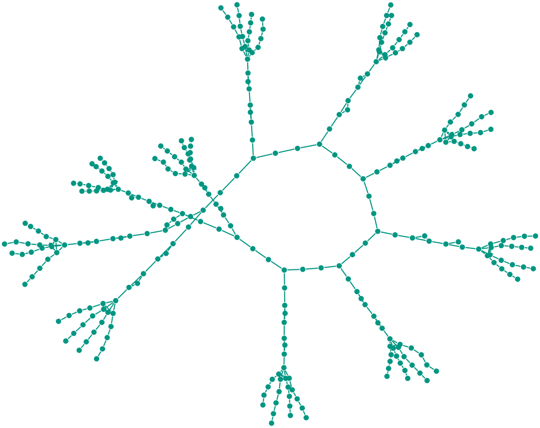
Karlsruhe Institute of Technology



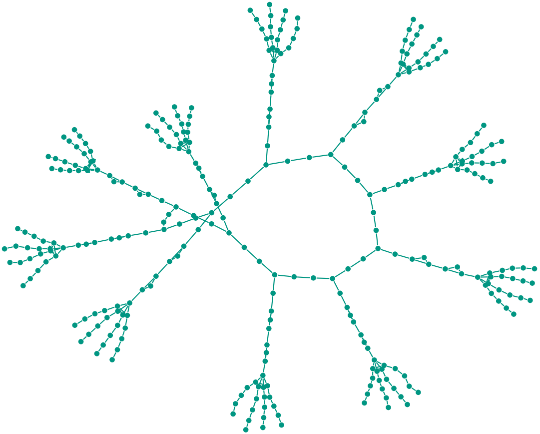
Layout Interpolation (XLINEAR)



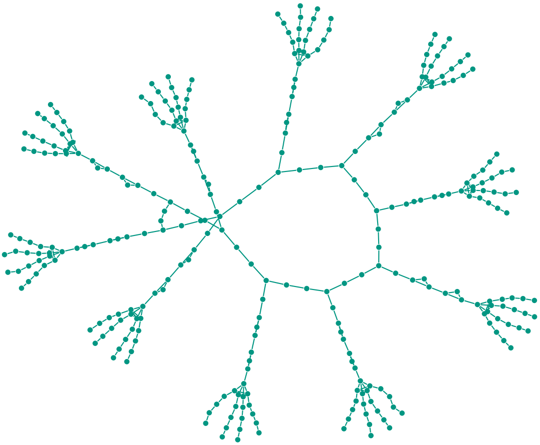
Layout Interpolation (XLINEAR)



Layout Interpolation (XLINEAR)



Layout Interpolation (XLINEAR)



Contents

Introduction

Methodology

Statistical Syndromes

Data Generation

Data Augmentation

Feature Extraction
- Entropy of Histograms
- Entropy of Sliding Averages

Discriminator Model

Evaluation

Conclusions & Future Work

Bibliography

Feature Extraction

Karlsruhe Institute of Technology

- Properties are multisets of unbounded size
- We need to condense them into a fixed-size feature vector
- Use histograms to present data
- Use sliding averages (Gaussian kernel) for RDF_LOCAL instead
- For each Property:
  - Arithmetic mean and root mean squared (2 values)
  - Entropy regression of histograms (2 values)
  - Differential entropy in case of RDF_LOCAL (1 value)
- Principal components (4 values)
- (Logarithm of) number of vertices and edges (2 values)

All features are normalized by subtracting the mean and dividing by the standard deviation.

Feature Extraction

- Properties are multisets of unbounded size
- We need to condense them into a fixed-size feature vector
- Use histograms t
- Use sliding avera ... L instead
- For each Property
  - Arithmetic mea
  - Entropy regres
  - Differential ent
- Principal compor
- (Logarithm of) nu

Sliding average with kernel f:

$$F_f(x) = \frac{\sum_{i=1}^{n} f(x, x_i)}{\int_{-\infty}^{+\infty} \mathrm{d}y \ \sum_{i=1}^{n} f(y, x_i)}$$

Gaussian kernel:

$$g_\sigma(\mu, x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

All features are normalized by subtracting the mean and dividing by the standard deviation.

Feature Extraction

Karlsruhe Institute of Technology

- Properties are multisets of unbounded size
- We need to condense them into a fixed-size feature vector
- Use histograms to present data
- Use sliding averages (Gaussian kernel) for RDF_LOCAL instead
- For each Property:
  - Arithmetic mean and root mean squared (2 values)
  - Entropy regression of histograms (2 values)
  - Differential entropy in case of RDF_LOCAL (1 value)
- Principal components (4 values)
- (Logarithm of) number of vertices and edges (2 values)

All features are normalized by subtracting the mean and dividing by the standard deviation.

Entropy of Histograms

KIT Karlsruhe Institute of Technology

Definition (Discrete Entropy)

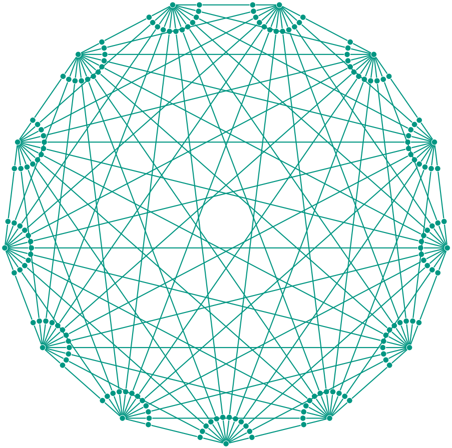
Let H be a histogram with $n \in \mathbb{N}$ bins that have the values (relative frequency counts) $H_1, \ldots, H_2 \in \mathbb{R}_{\geq 0}$ such that $\sum_{i=1}^{n} H_i = 1$. Then the entropy of H is

$$S(H) = -\sum_{i=1}^{n} H_i \log_2(H_i)$$

where we use the convention that bins with $H_i = 0$ shall contribute a zero term to the sum.

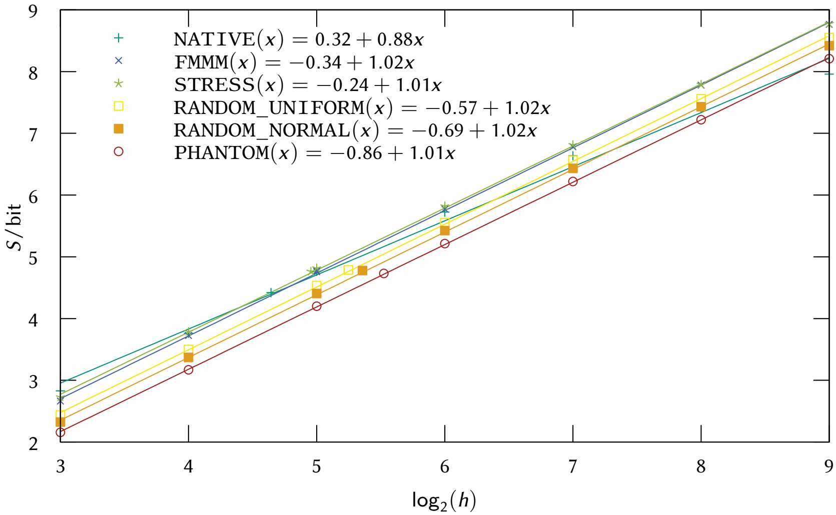
- Depends strongly on the bin width / count
- Entropy grows exponentially with bin count

Entropy of Histogram



Entropy of Histogram

Karlsruhe Institute of Technology



Entropy of Sliding Averages

Definition (Differential Entropy)

Let $f : \mathbb{R} \to \mathbb{R}_{\geq 0}$ be a non-negative steady function normalized such that $\int_{-\infty}^{+\infty} \mathrm{d}x \, f(x) = 1$. The *differential entropy* of f is defined as

$$\bar{S}(f) = -\int_{-\infty}^{+\infty} \mathrm{d}x \; x \log_2(x)$$

where we use the convention that the integrand shall be zero for those $x \in \mathbb{R}$ where $f(x) = 0$.

- Originally proposed by Shannon (1948)
- Not a measure of information (Jaynes 1963)
- Can actually be negative
- Remotely useful transcendental properties (Cover and Thomas 1991)

Contents

Introduction

Methodology

Statistical Syndromes

Data Generation

Data Augmentation

Feature Extraction

Discriminator Model

Evaluation

Conclusions & Future Work

Bibliography

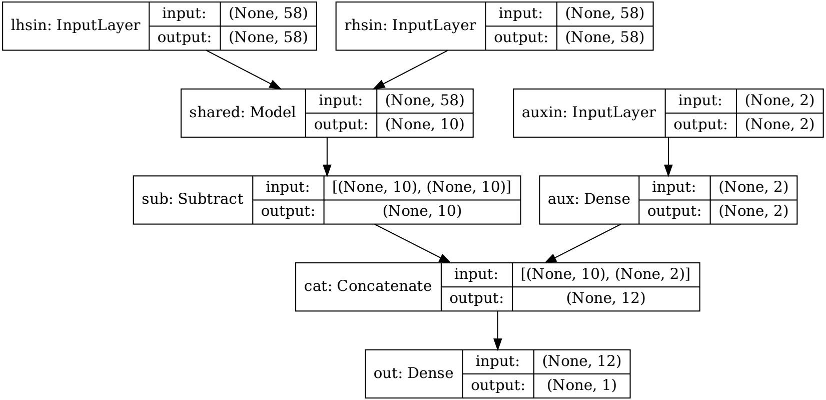
Discriminator Model

- Receives two feature vectors of layouts Γ_A and Γ_B as inputs
- Outputs a number between -1 and $+1$

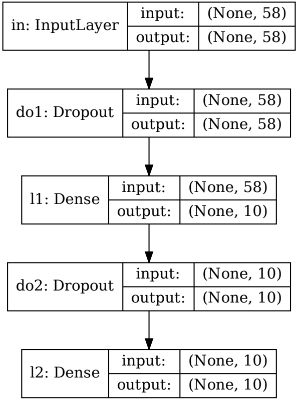
$$\mathcal{D}(\Gamma_A, \Gamma_B) = \begin{cases} < 0, & \Gamma_A \text{ is considered better} \\ > 0, & \Gamma_B \text{ is considered better} \\ = 0, & \text{neither layout is considered better} \end{cases}$$

- Siamese neural network (Bromley et al. 1994)

Discriminator Model



Discriminator Model



Contents

Introduction

Methodology

Statistical Syndromes

Data Generation

Data Augmentation

Feature Extraction

Discriminator Model

Evaluation

- Accuracy
- Contribution of Individual Properties

Conclusions & Future Work

Bibliography

Evaluation

- Corpora with tens of thousands of labeled pairs $(\Gamma_\mathrm{A}, \Gamma_\mathrm{B}, t)$
- We ensure that $t \not\approx 0$ (*i.e.* we don't use borderline cases)
- 20 % (chosen randomly) of this data is set aside for testing
- $\mathcal{D}(\Gamma_\mathrm{A}, \Gamma_\mathrm{B}) = p$ is considered a success if and only if $\mathrm{sign}(p) = \mathrm{sign}(t)$
- Training and testing are repeated with different partitions (cross validation via random subsampling)
- Reproducible success rates in excess of 95 % achieved

Accuracy

	Cond. Neg.		*Cond. Pos.*		Σ	
Pred. Neg.	48.31 %	± 0.70 %	1.25 %	± 0.60 %	49.56 %	± 1.17 %
Pred. Pos.	1.81 %	± 0.65 %	48.63 %	± 0.68 %	50.44 %	± 1.17 %
Σ	50.12 %	± 0.60 %	49.88 %	± 0.60 %	100.00 %	± 0.00 %

Success Rate:	96.94 %	± 0.12 %
Failure Rate:	3.06 %	± 0.12 %
Average Number of Tests:		≈ 11 762
Number of Repetitions:		10

Contribution of Individual Properties

Karlsruhe Institute of Technology

Property	*Sole Exclusion*		*Sole Inclusion*	
RDF_LOCAL	88.77 %	±5.34 %	96.21 %	± 0.37 %
PRINCOMP2ND	96.69 %	±0.24 %	58.27 %	± 3.55 %
EDGE_LENGTH	96.85 %	±0.20 %	71.36 %	±10.07 %
ANGULAR	96.88 %	±0.20 %	85.70 %	± 6.19 %
RDF_GLOBAL	96.91 %	±0.30 %	88.09 %	± 1.64 %
TENSION	96.96 %	±0.24 %	92.07 %	± 0.22 %
PRINCOMP1ST	97.12 %	±0.16 %	62.91 %	± 8.70 %
Baseline Using All Properties	96.94 %	±0.12 %		

Contents

Introduction

Methodology

Statistical Syndromes

Data Generation

Data Augmentation

Feature Extraction

Discriminator Model

Evaluation

Conclusions & Future Work

- Summary
- Open Questions and Future Plans

Bibliography

Summary

- Built framework for flexible experimentation
- Developed several graph generators
- Explored ways of data augmentation
- Investigated several properties (some of them promising)
- RDF_LOCAL is most valuable but also most expensive
- Demonstrated feasibility by training neural network
- All experiments run fully automatic and can be repeated by anybody who wishes to do so
- Download source code at `http://klammler.eu/msc/`

Open Questions and Future Plans

Karlsruhe Institute of Technology

- Additional Properties
  - Properties involving *edges*
  - Shapelet analysis
  - More correlations between graph-theoretical and layout properties
- More elaborate data analysis
- Comparison with existing measures
- User study
- Extension to more general graph drawings
- Application as a meta-heuristic for a genetic layout algorithm

Contents

Introduction

Methodology

Statistical Syndromes

Data Generation

Data Augmentation

Feature Extraction

Discriminator Model

Evaluation

Conclusions & Future Work

Bibliography

Bibliography I

Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; Shah, R. In *Advances in Neural Information Processing Systems*, 1994, pp 737–744.

Cover, T. M.; Thomas, J. A., *Elements of information theory*, Wiley series in telecommunications; Wiley: 1991.

Huang, W.; Huang, M. L.; Lin, C.-C. *Information Sciences* **2016**, *330*, 444–454.

Jaynes, E. T. In *Statistical Physics*, Ford, K., Ed.; Brandeis University Summer Institute Lectures in Theoretical Physics 3; Benjamin: New York, 1963, pp 181–218.

Kamada, T.; Kawai, S. *Information Processing Letters* **1989**, *31*, 7–15.

Klapaukh, R. An Empirical Evaluation of Force-Directed Graph Layout., Ph.D. Thesis, Victoria University of Wellington, 2014.

Bibliography II

Koren, Y.; Çivril, A. In *International Symposium on Graph Drawing*, Springer: 2008, pp 193–205.

Purchase, H. *Journal of Visual Languages and Computing* **2002**, *13*, 501–516.

Schaefer, S.; McPhail, T.; Warren, J. In *ACM transactions on graphics (TOG)*, 2006; Vol. 25, pp 533–540.

Shannon, C. E. *The Bell System Technical Journal* **1948-10**, *27*, 623–656.

Please refer to the printed thesis for a complete list of references.