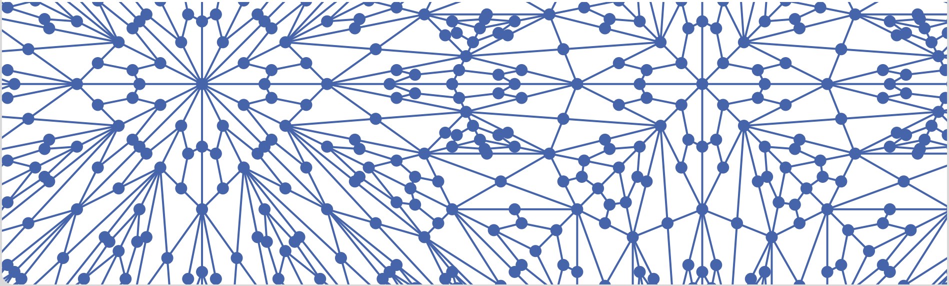
Karlsruhe Institute of Technology

Aesthetic Discrimination of Graph Layouts

Moritz Klammler · Tamara Mchedlidze · Alexey Pak

26th International Symposium on Graph Drawing and Network Visualization, Barcelona (2018)



Abstract

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

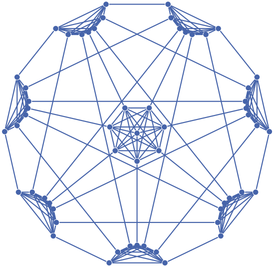
Klammler, M. et al. Aesthetic Discrimination of Graph Layouts., 2018

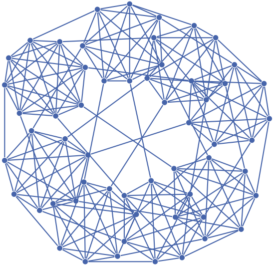
Klammler, M. Aesthetic value of graph layouts: Investigation of statistical syndromes for automatic quantification., Master's thesis, Karlsruhe Institute of Technology, 2018

Klammler, M. et al. Source Code for Aesthetic Discrimination of Graph Layouts., 2018

Problem Statement

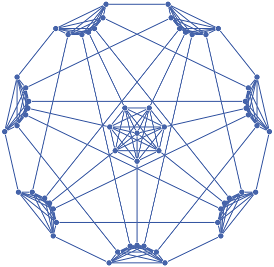
Given two vertex layouts Γ_a and Γ_b for the same simple graph $G = (V, E)$.
Is Γ_a or Γ_b more aesthetically pleasing?

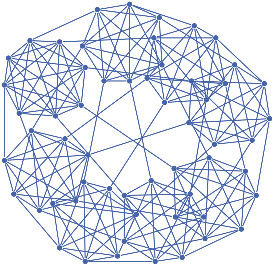




Problem Statement

Given two **vertex layouts** Γ_a and Γ_b for the same simple graph $G = (V, E)$.
Is Γ_a or Γ_b more aesthetically pleasing?

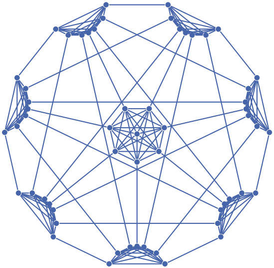


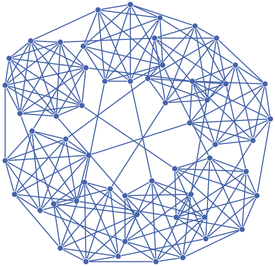


$$\begin{array}{rccc} \Gamma: & V & \rightarrow & \mathbb{R}^2 \\ & v & \mapsto & (x_v, y_v) \end{array}$$

Problem Statement

Given two vertex layouts Γ_a and Γ_b for the same **simple graph** $G = (V, E)$.
Is Γ_a or Γ_b more aesthetically pleasing?

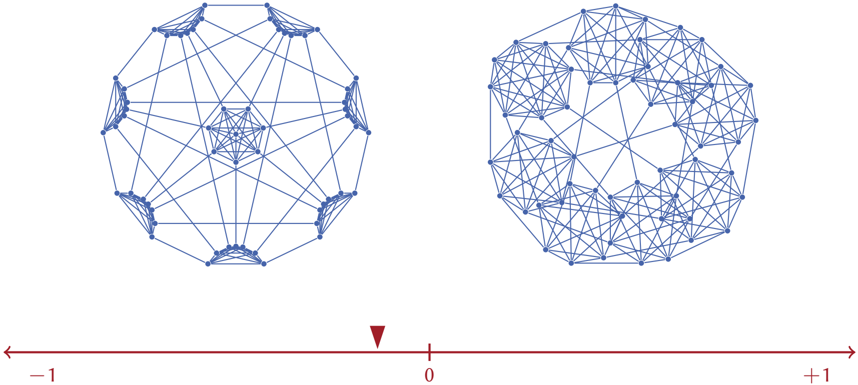




$\rightarrow$ undirected
$\rightarrow$ no loops
$\rightarrow$ no multiple edges

Problem Statement

Given two vertex layouts Γ_a and Γ_b for the same simple graph $G = (V, E)$.
Is Γ_a or Γ_b more aesthetically pleasing?



Contents

Related Work

Karlsruhe Institute of Technology

- Simple Metrics
  - number of edge crossings
  - minimum crossing angle (cross resolution)
  - minimum angle between incident edges (angular resolution)
  - standard deviation of edge lengths
  - ...
- $\mathrm{COMB}(\Gamma_i) = \sum_M w_M z_M(\Gamma_i)$ with $z_M = (M(\Gamma_i) - \mu_M) / \sigma_M$
- $\mathrm{STRESS}(\Gamma) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} k_{ij} \Big(\mathrm{dist}_\Gamma(v_i, v_j) - L \cdot \mathrm{dist}_G(v_i, v_j) \Big)^2$

Huang, W. et al. *J Vis Lang Comput* **2013**, *24*, 262–272

Kamada, T.; Kawai, S. *Inf Process Lett* **1989**, *31*, 7–15

Related Work

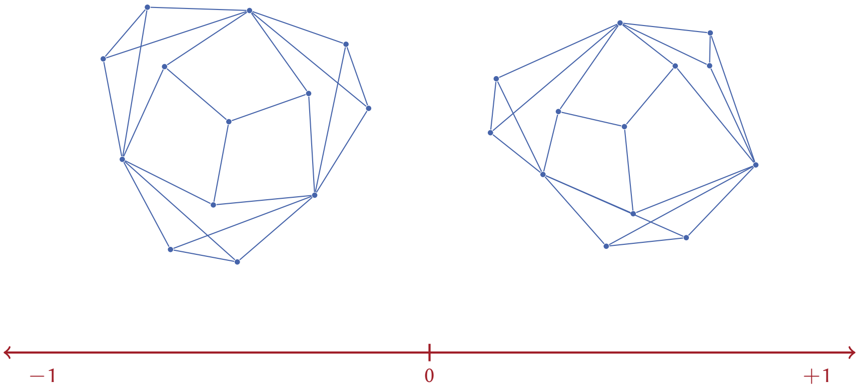
- Simple Metrics
  - number of edge crossings
  - minimum crossing angle (cross resolution)
  - minimum angle between incident edges (angular resolution)
  - standard deviation of edge lengths
  - ...
- $\textsc{comb}(\Gamma_i) = \sum_M w_M z_M(\Gamma_i)$ with $z_M = (M(\Gamma_i) - \mu_M) / \sigma_M$
- $\textsc{stress}(\Gamma) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} k_{ij} \left(\mathrm{dist}_\Gamma(v_i, v_j) - L \cdot \mathrm{dist}_G(v_i, v_j) \right)^2$

Huang, W. et al. *J Vis Lang Comput* **2013**, *24*, 262–272

Kamada, T.; Kawai, S. *Inf Process Lett* **1989**, *31*, 7–15

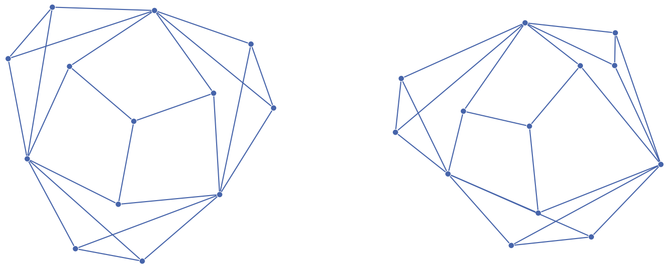
Related Work

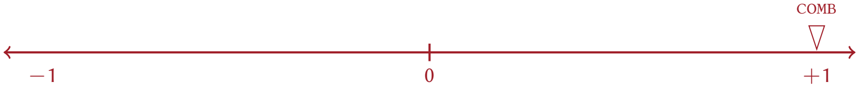
Combined Metric (COMB)



Related Work

Combined Metric (COMB)





Related Work

- Simple Metrics
  - number of edge crossings
  - minimum crossing angle (cross resolution)
  - minimum angle between incident edges (angular resolution)
  - standard deviation of edge lengths
  - ...
- $\textsc{Comb}(\Gamma_i) = \sum_M w_M z_M(\Gamma_i)$ with $z_M = (M(\Gamma_i) - \mu_M) / \sigma_M$
- $\textsc{Stress}(\Gamma) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} k_{ij} \left(\mathrm{dist}_\Gamma(v_i, v_j) - L \cdot \mathrm{dist}_G(v_i, v_j) \right)^2$

Huang, W. et al. *J Vis Lang Comput* **2013**, *24*, 262–272

Kamada, T.; Kawai, S. *Inf Process Lett* **1989**, *31*, 7–15

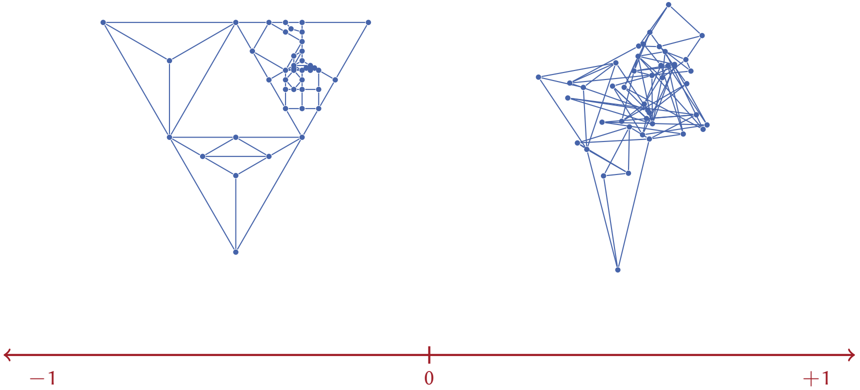
Related Work

- Simple Metrics
  - number of edge crossings
  - minimum crossing angle (cross resolution)
  - minimum angle between incident edges (angular resolution)
  - standard deviation of edge lengths
  - ...
- $\textsc{comb}(\Gamma_i) = \sum_M w_M\, z_M(\Gamma_i) \qquad \text{with} \qquad z_M = (M(\Gamma_i) - \mu_M) \,/\, \sigma_M$
- $\textsc{stress}(\Gamma) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} k_{ij} \Big(\mathrm{dist}_\Gamma(v_i, v_j) - L \cdot \mathrm{dist}_G(v_i, v_j) \Big)^2$

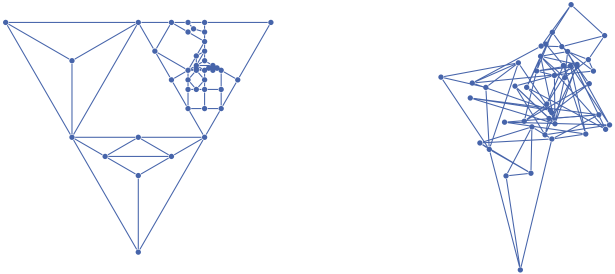
Huang, W. et al. *J Vis Lang Comput* **2013**, *24*, 262–272

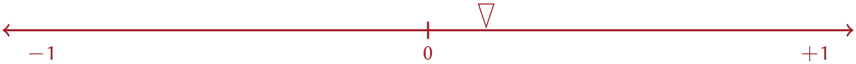
Kamada, T.; Kawai, S. *Inf Process Lett* **1989**, *31*, 7–15

Stress (STRESS)



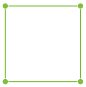
Stress (STRESS)



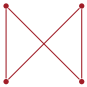


Contents

Methodology Overview

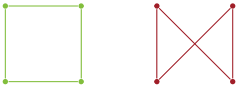


Γ_a

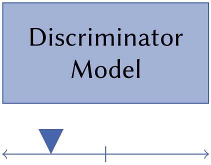


Γ_b

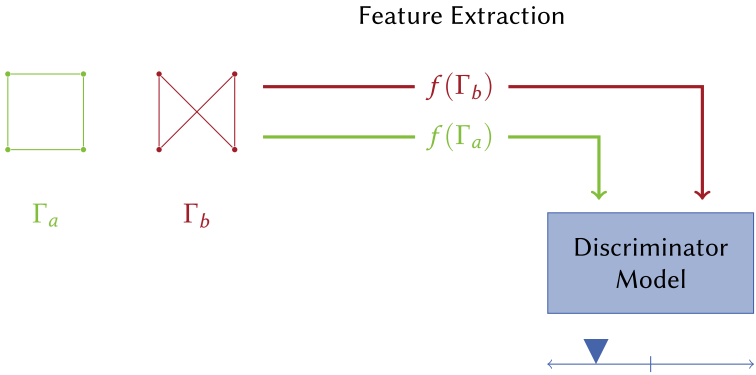
Methodology Overview



Γ_a Γ_b



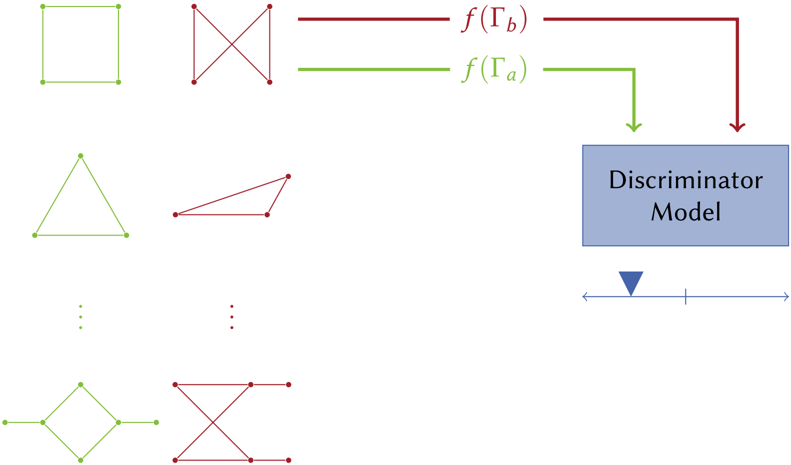
Methodology Overview



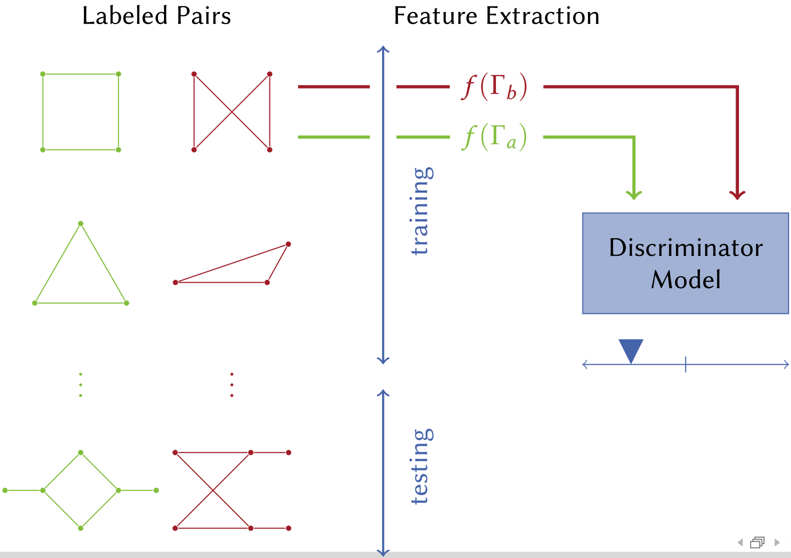
Methodology Overview

Labeled Pairs

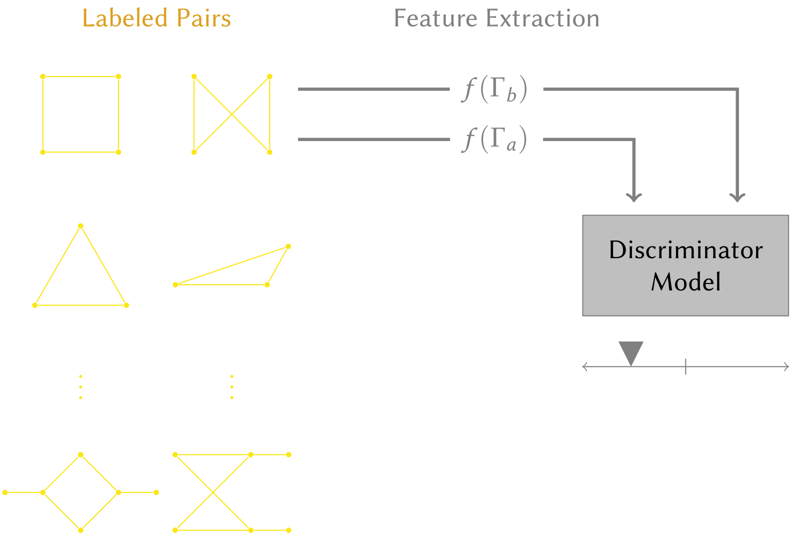
Feature Extraction



Methodology Overview



Methodology Overview



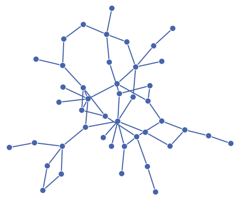
Data Acquisition & Augmentation

Karlsruhe Institute of Technology

1. Collect a large number of graphs
   - **imported graphs** — from public collections
   - **generated graphs** — using probabilistic algorithms
2. Compute various layouts for these graphs
   - **native layouts** — fall out of the graph generation process
   - **proper layouts** — using common state-of-the-art algorithms
   - **garbage layouts** — using more or less random placements of vertices
3. Use data augmentation to expand this corpus
   - **layout worsening** — degrade proper layout by given rate
   - **layout interpolation** — compute layout “between” proper and garbage layout

Data Acquisition & Augmentation

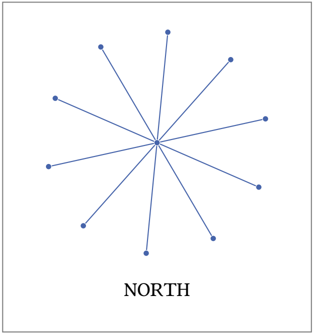
Imported Graphs

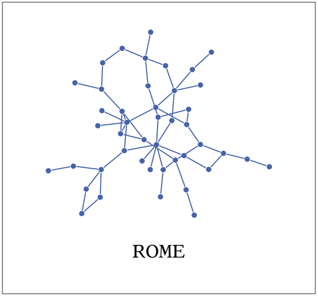


ROME

Data Acquisition & Augmentation

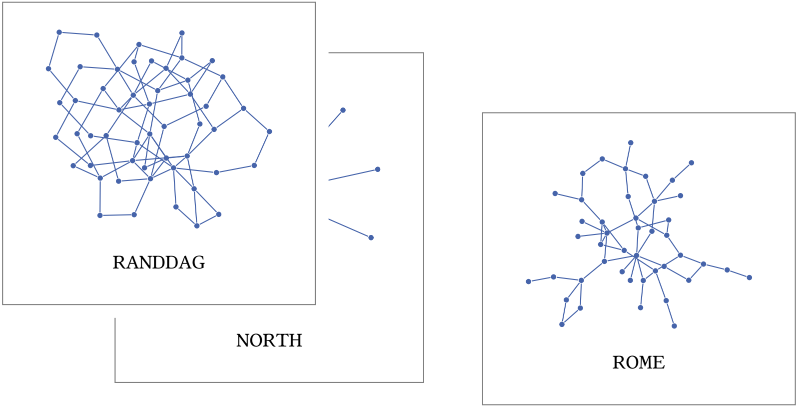
Imported Graphs





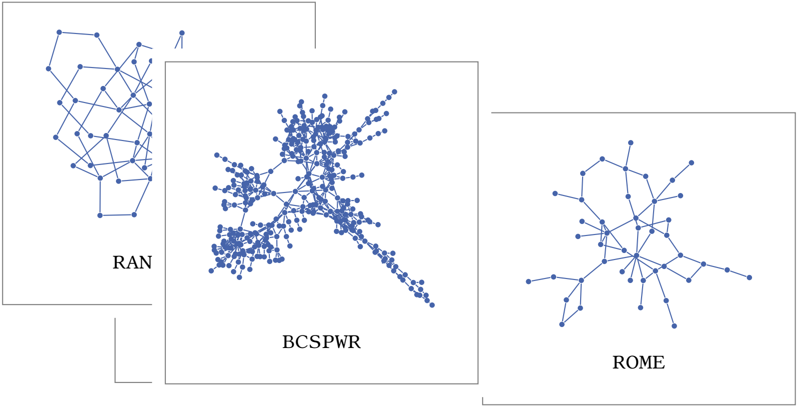
Data Acquisition & Augmentation

Imported Graphs



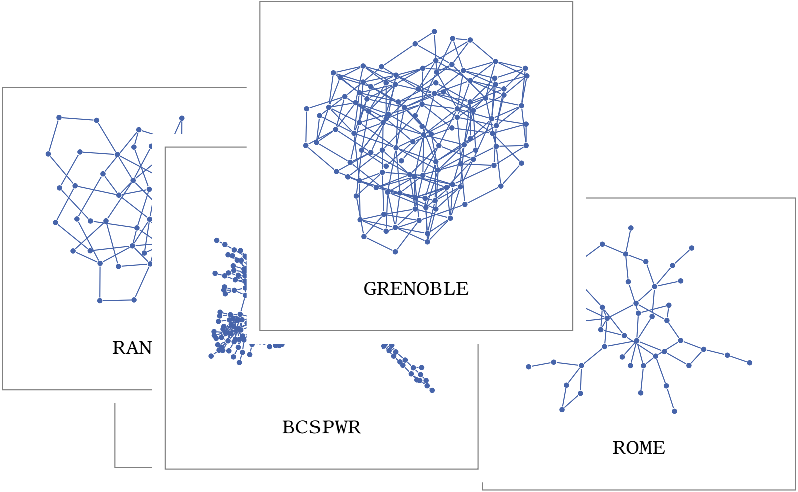
Data Acquisition & Augmentation

Imported Graphs



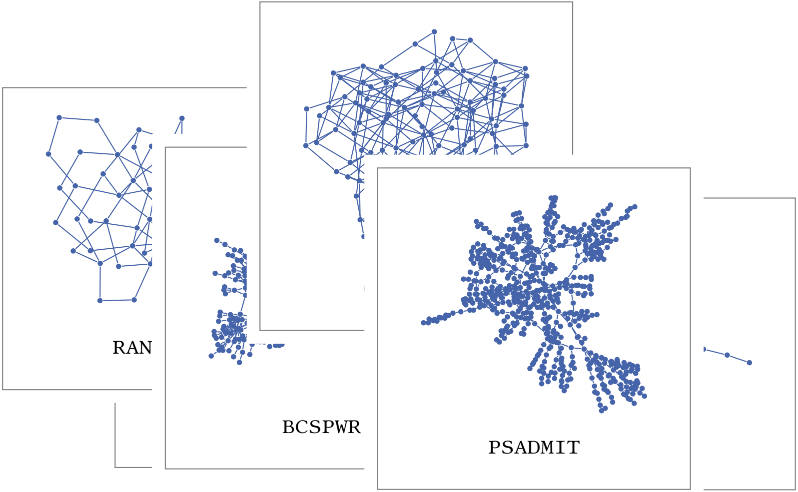
Data Acquisition & Augmentation

Imported Graphs



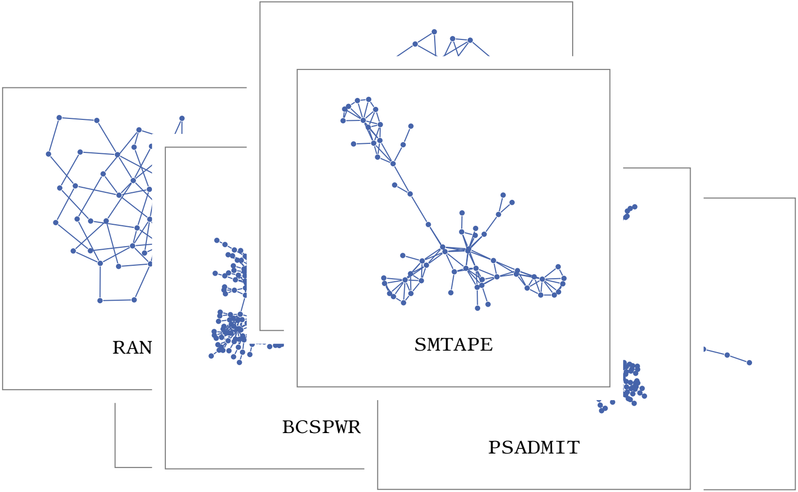
Data Acquisition & Augmentation

Imported Graphs



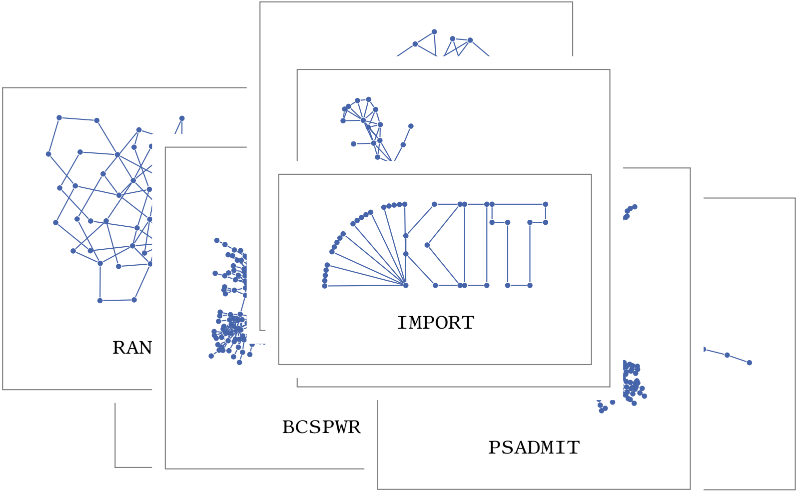
Data Acquisition & Augmentation

Imported Graphs



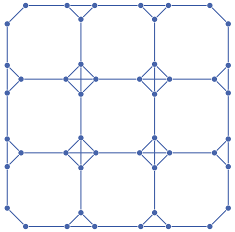
Data Acquisition & Augmentation

Imported Graphs



Data Acquisition & Augmentation

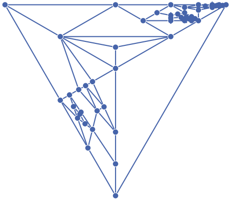
Generated Graphs



LINDENMAYER

Data Acquisition & Augmentation

Generated Graphs

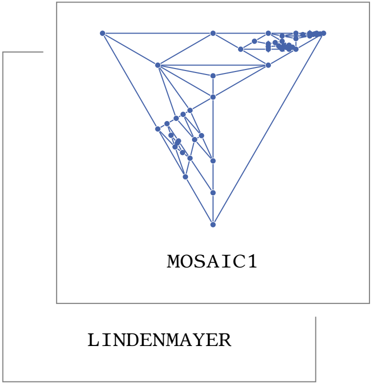


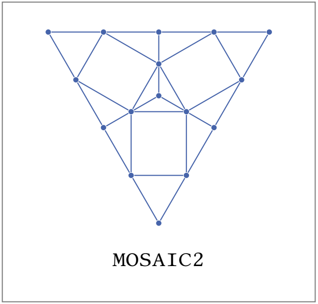
MOSAIC1

LINDENMAYER

Data Acquisition & Augmentation

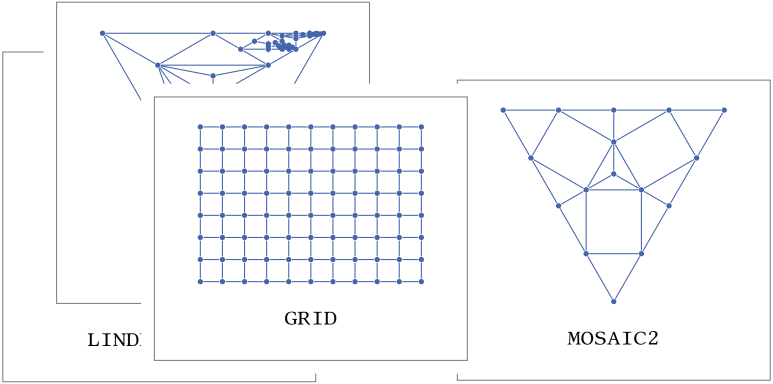
Generated Graphs





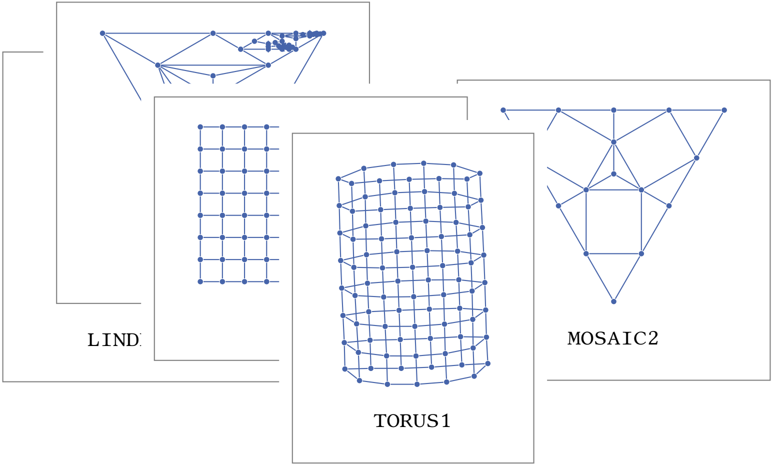
Data Acquisition & Augmentation

Generated Graphs



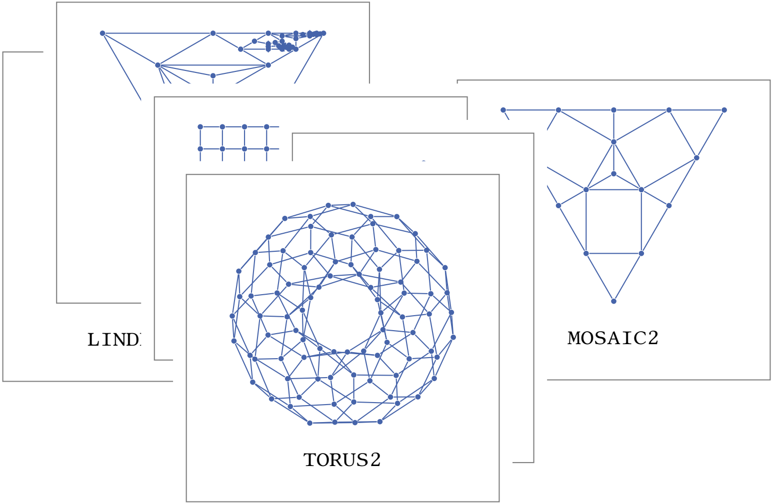
Data Acquisition & Augmentation

Generated Graphs



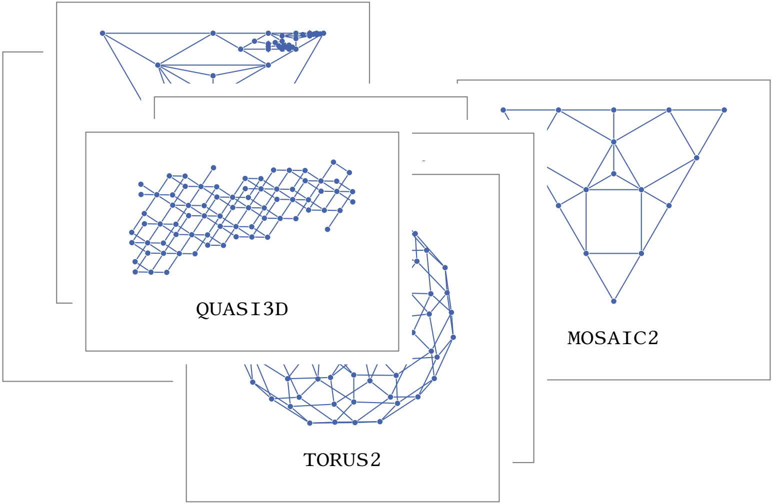
Data Acquisition & Augmentation

Generated Graphs



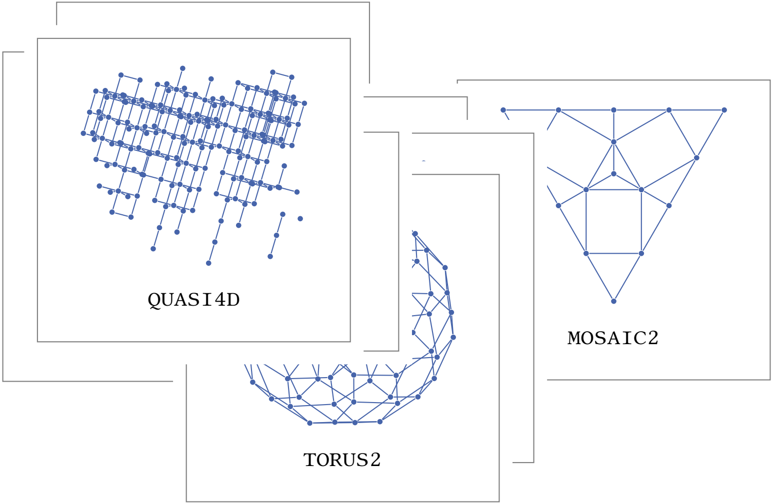
Data Acquisition & Augmentation

Generated Graphs



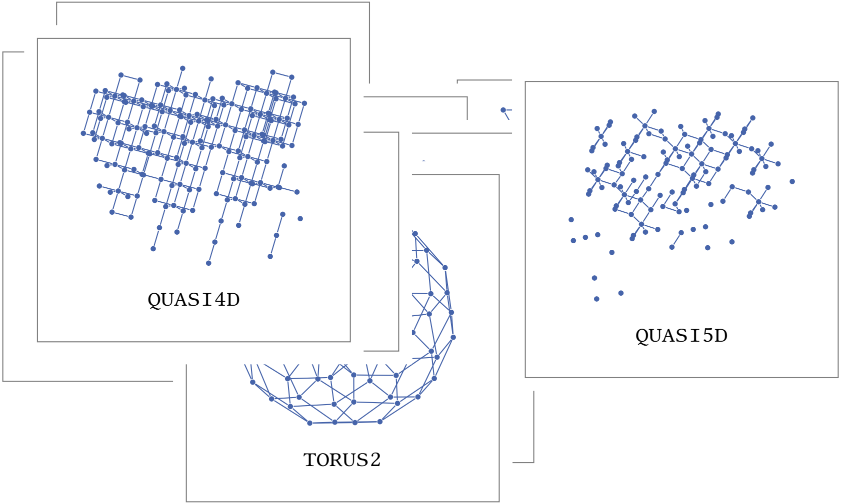
Data Acquisition & Augmentation

Generated Graphs



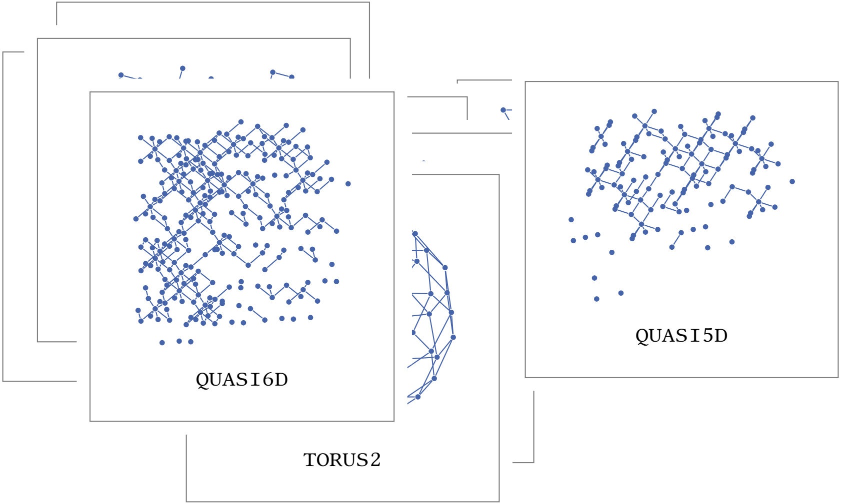
Data Acquisition & Augmentation

Generated Graphs



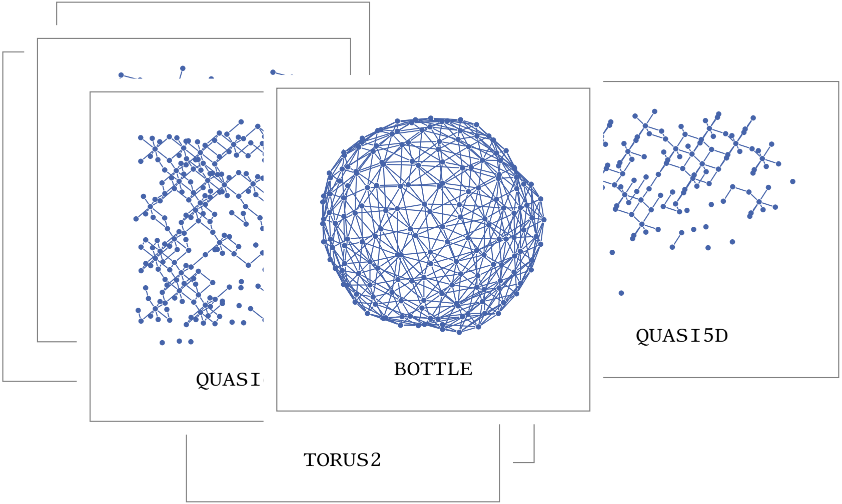
Data Acquisition & Augmentation

Generated Graphs



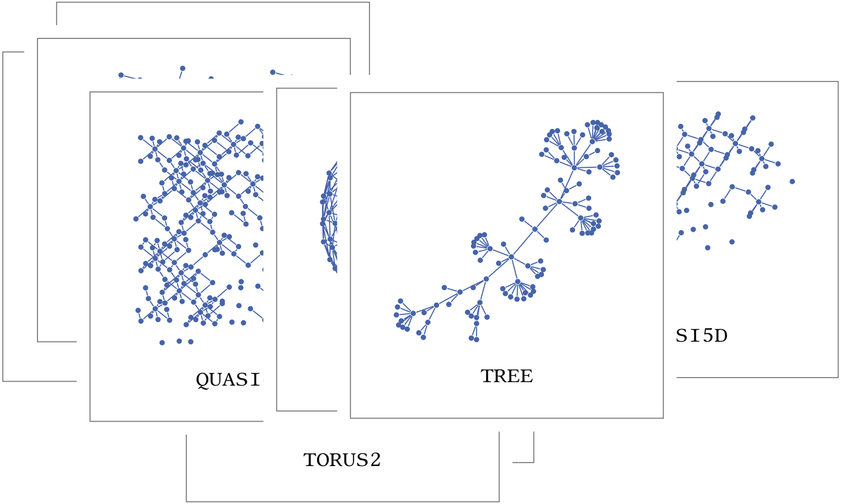
Data Acquisition & Augmentation

Generated Graphs



Data Acquisition & Augmentation

Generated Graphs



Data Acquisition & Augmentation

Karlsruhe Institute of Technology

1. Collect a large number of graphs
   - **imported graphs** — from public collections
   - **generated graphs** — using probabilistic algorithms
2. Compute various layouts for these graphs
   - **native layouts** — fall out of the graph generation process
   - **proper layouts** — using common state-of-the-art algorithms
   - **garbage layouts** — using more or less random placements of vertices
3. Use data augmentation to expand this corpus
   - **layout worsening** — degrade proper layout by given rate
   - **layout interpolation** — compute layout “between” proper and garbage layout

Data Acquisition & Augmentation

Karlsruhe Institute of Technology

1. Collect a large number of graphs
   - **imported graphs** — from public collections
   - **generated graphs** — using probabilistic algorithms

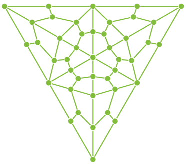
2. Compute various layouts for these graphs
   - **native layouts** — fall out of the graph generation process
   - **proper layouts** — using common state-of-the-art algorithms
   - **garbage layouts** — using more or less random placements of vertices

3. Use data augmentation to expand this corpus
   - **layout worsening** — degrade proper layout by given rate
   - **layout interpolation** — compute layout "between" proper and garbage layout

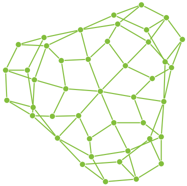
Data Acquisition & Augmentation

Layouts

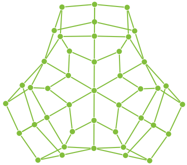
proper



NATIVE

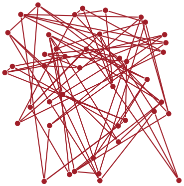


FMMM

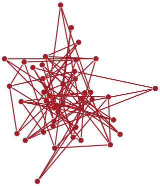


STRESS

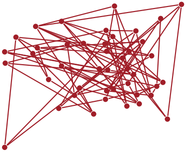
garbage



RANDOM_UNIFORM



RANDOM_NORMAL



PHANTOM

Data Acquisition & Augmentation

Karlsruhe Institute of Technology

1. Collect a large number of graphs
   - **imported graphs** — from public collections
   - **generated graphs** — using probabilistic algorithms

2. Compute various layouts for these graphs
   - **native layouts** — fall out of the graph generation process
   - **proper layouts** — using common state-of-the-art algorithms
   - **garbage layouts** — using more or less random placements of vertices

3. Use data augmentation to expand this corpus
   - **layout worsening** — degrade proper layout by given rate
   - **layout interpolation** — compute layout “between” proper and garbage layout

Data Acquisition & Augmentation

Karlsruhe Institute of Technology

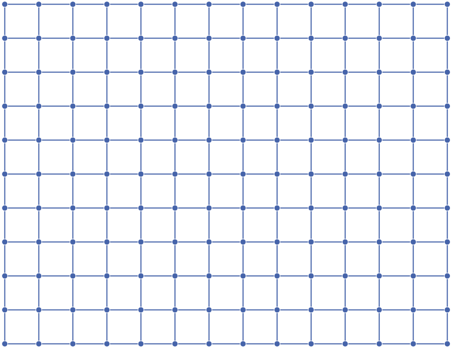
1. Collect a large number of graphs
   - **imported graphs** — from public collections
   - **generated graphs** — using probabilistic algorithms

2. Compute various layouts for these graphs
   - **native layouts** — fall out of the graph generation process
   - **proper layouts** — using common state-of-the-art algorithms
   - **garbage layouts** — using more or less random placements of vertices

3. Use data augmentation to expand this corpus
   - **layout worsening** — degrade proper layout by given rate
   - **layout interpolation** — compute layout "between" proper and garbage layout

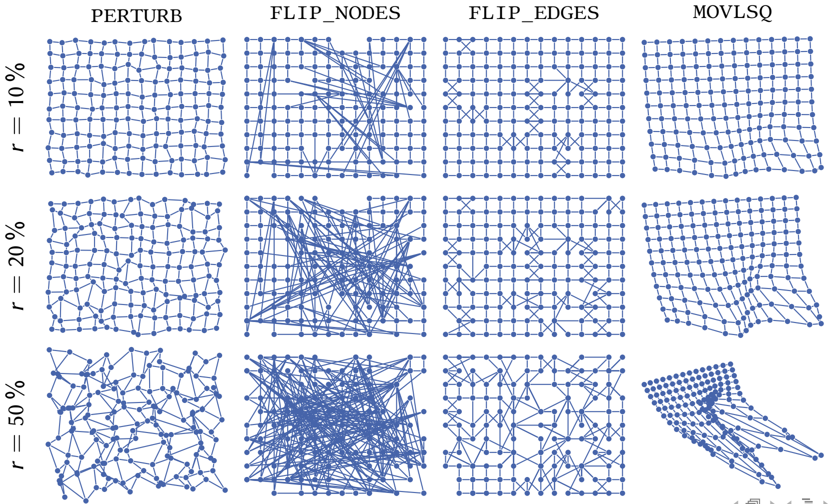
Data Acquisition & Augmentation

Layout Worsening



Data Acquisition & Augmentation

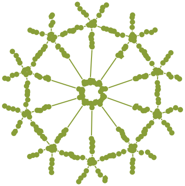
Layout Worsening



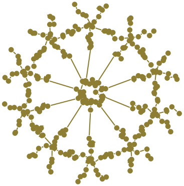
Data Acquisition & Augmentation

Layout Interpolation

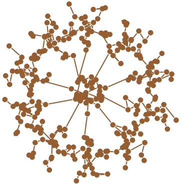
$r = 0\,\%$



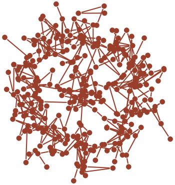
$r = 20\,\%$



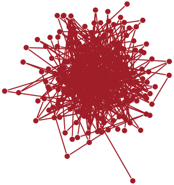
$r = 40\,\%$



$r = 60\,\%$

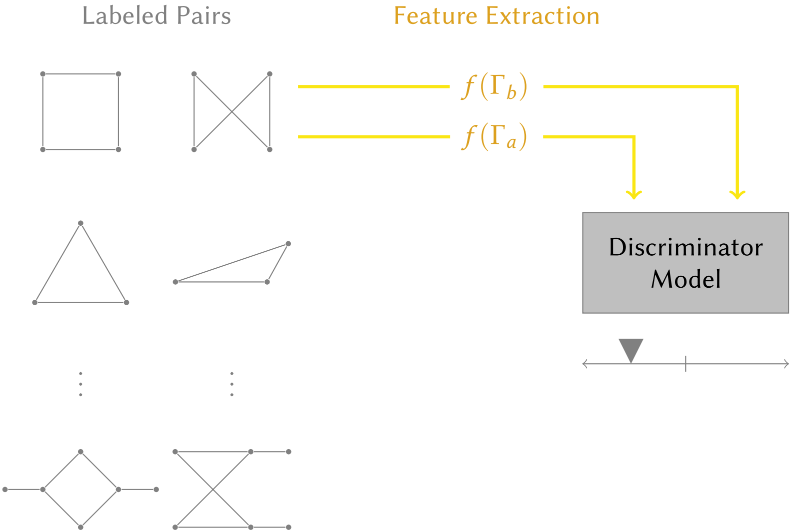


$r = 80\,\%$

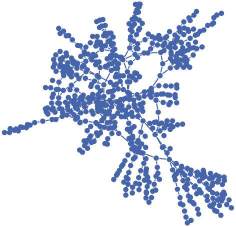


$r = 100\,\%$

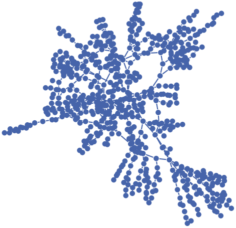
Methodology Overview



Feature Extraction Overview

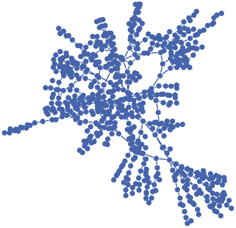


Feature Extraction Overview



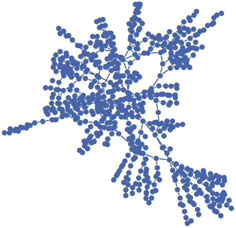
- PRINCOMP1
- PRINCOMP2
- ANGULAR
- EDGE_LENGTH
- TENSION
- RDF_GLOBAL
- RDF_LOCAL(d)

Feature Extraction Overview



PRINCOMP1	$\{x_1, x_2, \ldots\ldots\}$
PRINCOMP2	$\{x_1, x_2, \ldots\ldots\}$
ANGULAR	$\{x_1, x_2, \ldots\}$
EDGE_LENGTH	$\{x_1, x_2, \ldots\}$
TENSION	$\{x_1, x_2, \ldots\}$
RDF_GLOBAL	$\{x_1, x_2, \ldots\ldots\ldots\}$
RDF_LOCAL(d)	$\{x_1, x_2, \ldots\ldots\ldots\}(d)$

Feature Extraction Overview



PRINCOMP1	$\{x_1, x_2, \ldots\ldots\}$
PRINCOMP2	$\{x_1, x_2, \ldots\ldots\}$
ANGULAR	$\{x_1, x_2, \ldots\}$
EDGE_LENGTH	$\{x_1, x_2, \ldots\}$
TENSION	$\{x_1, x_2, \ldots\}$
RDF_GLOBAL	$\{x_1, x_2, \ldots\ldots\ldots\}$
RDF_LOCAL(d)	$\{x_1, x_2, \ldots\ldots\ldots\}(d)$

Statistical Syndromes of Graph Layouts

- `PRINVEC1` and `PRINVEC2` — first and second principal axis of vertex coordinates
- `PRINCOMP1` and `PRINCOMP2` — projections of vertex coordinates onto principal axes
- `ANGULAR` — angles between incident edges
- `EDGE_LENGTH` — edge lengths
- `TENSION` – ratios of Euclidean and graph-theoretical distances computed for all vertex pairs
- `RDF_GLOBAL` — pairwise distances between vertex coordinates
- `RDF_LOCAL`(d) — pairwise distances between vertex coordinates where the graph-theoretical distance between them is bounded by $d \in \mathbb{N}$

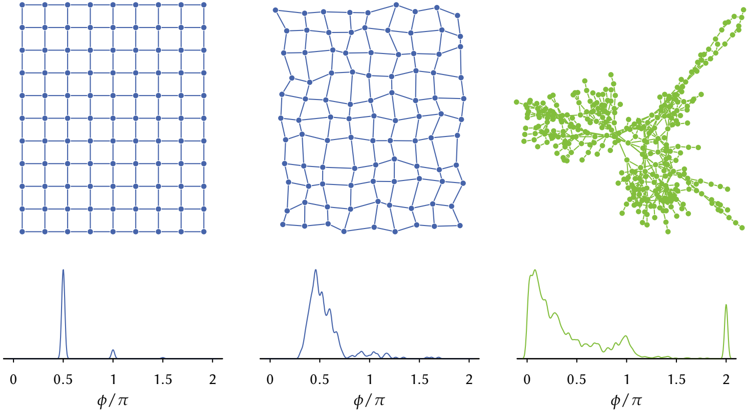
Statistical Syndromes of Graph Layouts

Karlsruhe Institute of Technology

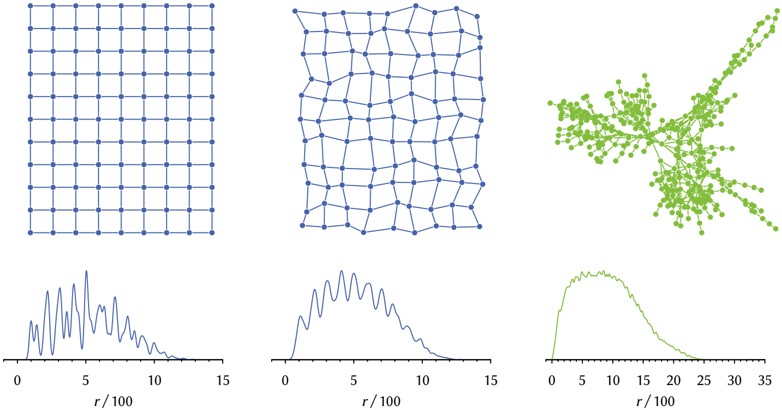
- PRINVEC1 and PRINVEC2 — first and second principal axis of vertex coordinates
- PRINCOMP1 and PRINCOMP2 — projections of vertex coordinates onto principal axes
- ANGULAR — angles between incident edges
- EDGE_LENGTH — edge lengths
- TENSION – ratios of Euclidean and graph-theoretical distances computed for all vertex pairs
- RDF_GLOBAL — pairwise distances between vertex coordinates
- RDF_LOCAL(d) — pairwise distances between vertex coordinates where the graph-theoretical distance between them is bounded by $d \in \mathbb{N}$

$$\begin{aligned} \text{RDF\_LOCAL}(\ 1\ ) &= \text{EDGE\_LENGTH} \\ \text{RDF\_LOCAL}(\infty) &= \text{RDF\_GLOBAL} \end{aligned}$$

Angles Between Incident Edges (ANGULAR)



Radial Distribution Function (RDF_GLOBAL)



Feature Extraction Overview

$$\text{PRINCOMP1} \quad \{x_1, x_2, \ldots\ldots\} \rightsquigarrow (y_1, y_2, y_3, y_4, y_5, y_6)$$

$$\text{PRINCOMP2} \quad \{x_1, x_2, \ldots\ldots\} \rightsquigarrow (y_1, y_2, y_3, y_4, y_5, y_6)$$

$$\text{ANGULAR} \quad \{x_1, x_2, \ldots\} \rightsquigarrow (y_1, y_2, y_3, y_4)$$

$$\text{EDGE\_LENGTH} \quad \{x_1, x_2, \ldots\} \rightsquigarrow (y_1, y_2, y_3)$$

$$\text{TENSION} \quad \{x_1, x_2, \ldots\} \rightsquigarrow (y_1, y_2, y_3, y_4)$$

$$\text{RDF\_GLOBAL} \quad \{x_1, x_2, \ldots\ldots\ldots\} \rightsquigarrow (y_1, y_2, y_3, y_4)$$

$$\text{RDF\_LOCAL}(d) \quad \{x_1, x_2, \ldots\ldots\ldots\}(d) \rightsquigarrow (y_1, y_2, y_3)(d)$$

Feature Extraction Overview

PRINCOMP1 $\{x_1, x_2, \ldots\ldots\} \rightsquigarrow (y_1, y_2, y_3, y_4, y_5, y_6)$

PRINCOMP2 $\{x_1, x_2, \ldots\ldots\} \rightsquigarrow (y_1, y_2, y_3, y_4, y_5, y_6)$

ANGULAR $\{x_1, x_2, \ldots\} \rightsquigarrow (y_1, y_2, y_3, y_4)$

EDGE_LENGTH $\{x_1, x_2, \ldots\} \rightsquigarrow (y_1, y_2, y_3)$

TENSION $\{x_1, x_2, \ldots\} \rightsquigarrow (y_1, y_2, y_3, y_4)$

RDF_GLOBAL $\{x_1, x_2, \ldots\ldots\ldots\} \rightsquigarrow (y_1, y_2, y_3, y_4)$

RDF_LOCAL(d) $\{x_1, x_2, \ldots\ldots\ldots\}(d) \rightsquigarrow (y_1, y_2, y_3)(d)$

Feature Vector

Feature Extraction

Karlsruhe Institute of Technology

We need to condense syndromes into a feature vector of fixed size.

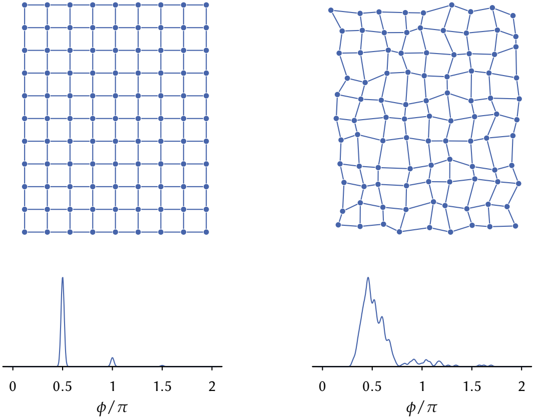
- arithmetic mean
- root mean squared (RMS)

- entropy of distribution
  - Problem: depends on data aggregation (histogram bin / filter width)
  - → Compute entropy for several histograms with different bin widths
  - → Perform linear regression
  - → Use regression parameters instead of entropy

Feature Extraction

We need

- arithr
- root r
- entro
  - Pr
  - →
  - →
  - →



Feature Extraction

We need to condense syndromes into a feature vector of fixed size.

- arithmetic mean
- root mean squared (RMS)

- entropy of distribution
  - Problem: depends on data aggregation (histogram bin / filter width)
  - $\rightarrow$ Compute entropy for several histograms with different bin widths
  - $\rightarrow$ Perform linear regression
  - $\rightarrow$ Use regression parameters instead of entropy

Feature Extraction

We need to condense syndromes into a feature vector of fixed size.

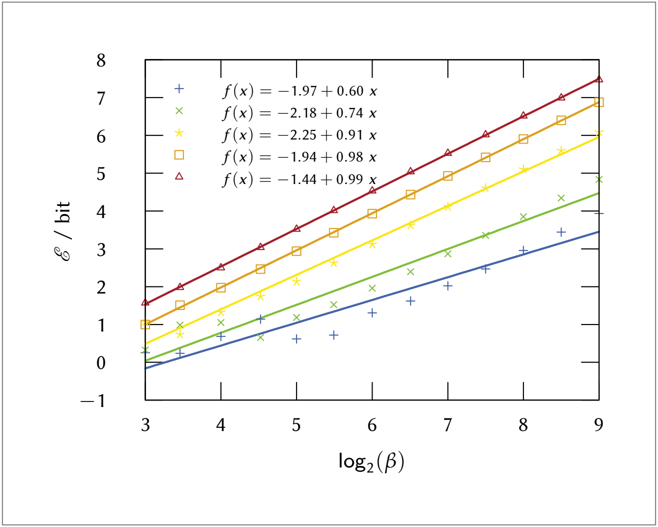
- arithmetic mean
- root mean squared (RMS)

- entropy of distribution
  - Problem: depends on data aggregation (histogram bin / filter width)
  - $\rightarrow$ Compute entropy for several histograms with different bin widths
  - $\rightarrow$ Perform linear regression
  - $\rightarrow$ Use regression parameters instead of entropy

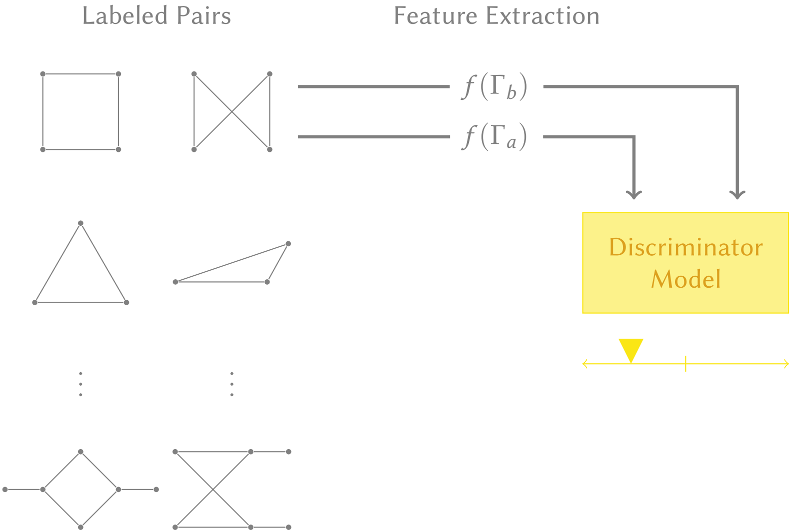
Feature Extraction

We need

- arithn
- root n
- entrop
  - Pr
  - →
  - →
  - →

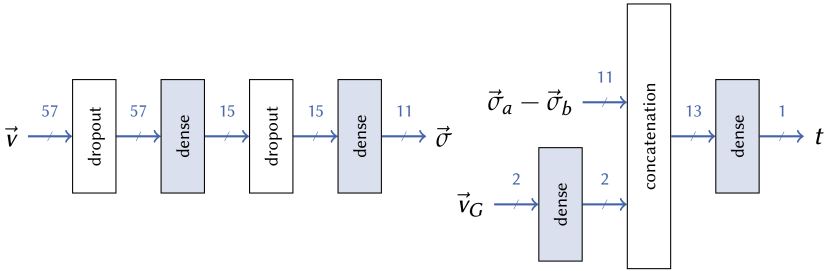


Methodology Overview



Siamese Neural Network

Karlsruhe Institute of Technology



Bromley, J. et al. *Adv Neural Inf Process Syst* **1994**, ed. by Jiang, X.; Wang, P. S. P., 737–744

Contents

Comparison With Other Metrics

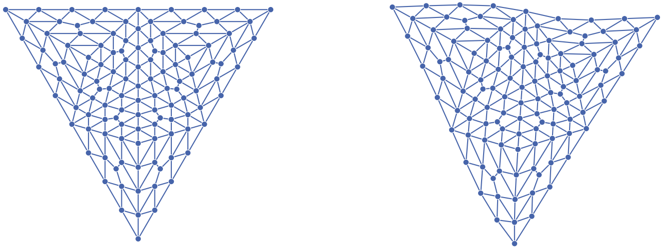
Metric	*Success Rate*	*Advantage*
DISC_MODEL	$(96.48 \pm 0.85)\,\%$	$(0.00 \pm 0.00)\,\%$
STRESS	$(93.49 \pm 0.86)\,\%$	$(2.99 \pm 1.01)\,\%$
COMB	$(92.76 \pm 1.03)\,\%$	$(3.71 \pm 1.22)\,\%$

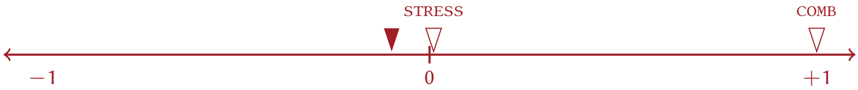
- 10-fold Cross validation via random sub-sampling
- STRESS was compared for best scale
- COMB weights were fitted to training data set

Welch, E.; Kobourov, S. *Comput Graph Forum* **2017**, *36*, 341–351

Huang, W. et al. *J Vis Lang Comput* **2013**, *24*, 262–272

Comparison With Other Metrics





Significance of Individual Syndromes

Property	*Sole Exclusion*	*Sole Inclusion*
PRINCOMP1	$(96.37 \pm 0.84)\,\%$	$(55.51 \pm 6.50)\,\%$
PRINCOMP2	$(96.20 \pm 0.76)\,\%$	$(61.08 \pm 5.24)\,\%$
EDGE_LENGTH	$(96.33 \pm 0.59)\,\%$	$(71.65 \pm 3.38)\,\%$
ANGULAR	$(96.40 \pm 0.34)\,\%$	$(77.79 \pm 6.06)\,\%$
RDF_GLOBAL	$(95.92 \pm 0.94)\,\%$	$(86.37 \pm 3.43)\,\%$
TENSION	$(96.83 \pm 0.31)\,\%$	$(89.78 \pm 0.95)\,\%$
RDF_LOCAL	$(90.04 \pm 2.04)\,\%$	$(94.78 \pm 1.60)\,\%$
Baseline Using All Properties	$(96.48 \pm 0.85)\,\%$	

- Note that RDF_LOCAL refers to a whole set of syndromes.

Contents

Conclusion and Future Work

Karlsruhe Institute of Technology

- Binary discrimination instead of absolute aesthetic measure
- Avoid a priori assumptions about influence on aesthetics
- Use of statistical syndromes inspired by Statistical Physics and Crystallography
- Data driven approach (machine learning)
- Accuracy usually outperforms other metrics
- `https://github.com/5gon12eder/msc-graphstudy`

- Identification of necessary and sufficient syndromes
- Optimization of the neural network
- Validation against human-labeled data

Bibliography

- Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; Shah, R. *Adv Neural Inf Process Syst* **1994**, ed. by Jiang, X.; Wang, P. S. P., 737–744.
- Huang, W.; Eades, P.; Hong, S.-H.; Lin, C.-C. *J Vis Lang Comput* **2013**, *24*, 262–272.
- Kamada, T.; Kawai, S. *Inf Process Lett* **1989**, *31*, 7–15.
- Klammler, M. Aesthetic value of graph layouts: Investigation of statistical syndromes for automatic quantification., Master's thesis, Karlsruhe Institute of Technology, 2018.
- Klammler, M. et al. Source Code for Aesthetic Discrimination of Graph Layouts., 2018.
- Klammler, M.; Mchedlidze, T.; Pak, A. Aesthetic Discrimination of Graph Layouts., 2018.
- Welch, E.; Kobourov, S. *Comput Graph Forum* **2017**, *36*, 341–351.