# Symmetry Detection and Classification in Drawings of Graphs


**Felice De Luca**

Iqbal Hossain

Stephen Kobourov

THE UNIVERSITY OF ARIZONA

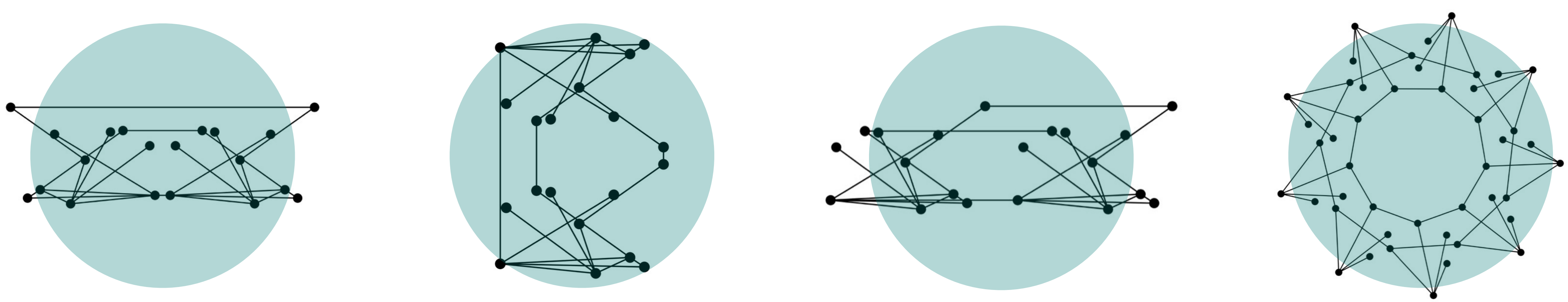
# Our contribution

A technique to **detect** and **classify** symmetries in GD using Convolutional Neural Network

1. **Is this layout symmetric?** yes, no.
2. **What type of symmetry does it show?** horizontal, vertical, translational, rotational.

An algorithm to **generate** symmetric graphs and symmetric layouts.

A **dataset** of symmetric drawings of graphs for machine learning projects



# Symmetries

Arcosanti (AZ)
experimental city

Tucson (AZ)

Sedona (AZ)
Chapel of the Holy Cross

Arizona Flowers
by Kathy Klein

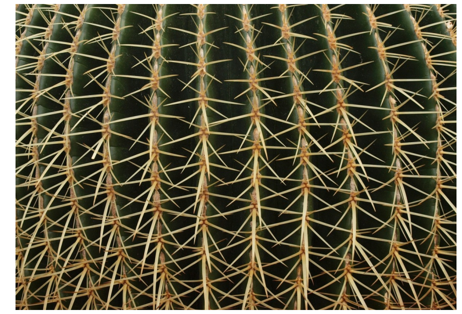
Scorpion in Sedona
(AZ)

Tarantula Hawk by
Ric Nielsen

# Symmetry types

Vertical

Horizontal



Translational

Rotational

*Reflectional Versions*: Reflection across an axis

- **Vertical**: vertical axis
- **Horizontal**: horizontal axis

*Translational Versions:* Repetition by shifting

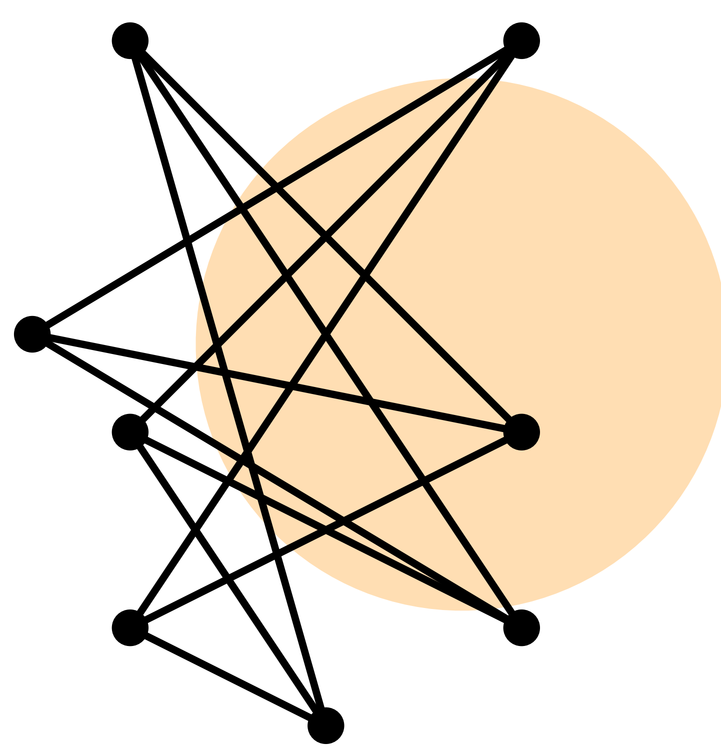
- **Translational**: parallel axes
- **Rotational**: radial axes

# Symmetry and abstract graphs

The symmetry of a graphs is known as automorphism
(Lubiw 1981)

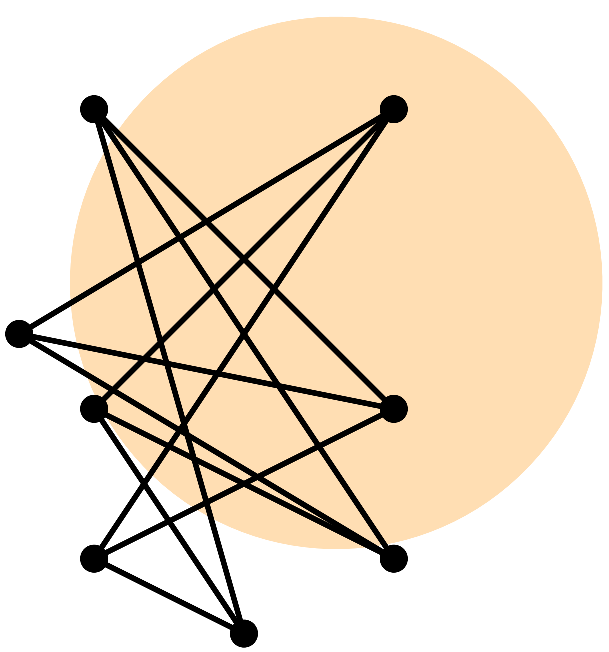
Detecting symmetries in an abstract graph is NP-complete
(Manning 1990)

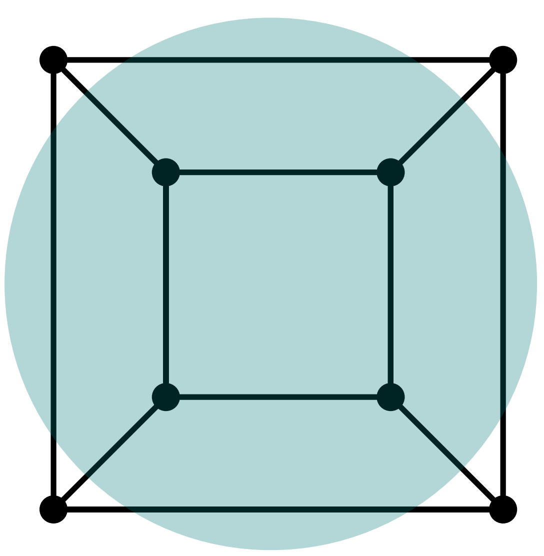
Mathematical heuristic to detect symmetries in graphs
(de Frasseix 1999)



# Symmetries and graph layout

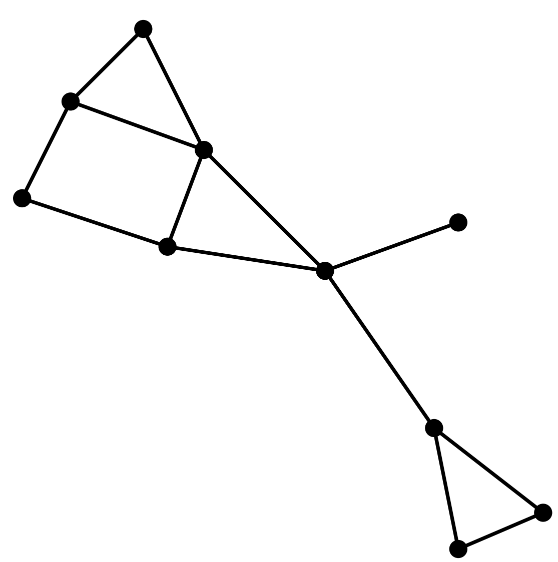
Symmetry is one of the most important aesthetic criteria that clearly reveals the structure and properties of a graph
(Eades and Hong 2013)





# Metrics for symmetries in graph layout

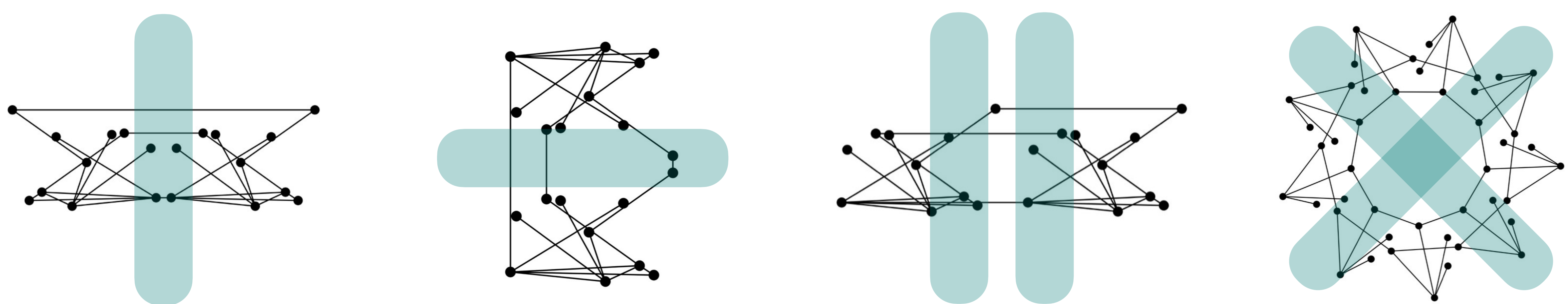
- Klapaukh (2014) and Purchase (2002) score symmetry in the range [0, 1]
- K and P are sensitive to the scaling of the layout
- Scores often disagree
- Human judgment agrees more often with P metric (Welsh and Kobourov 2017)



| | |
|---|---|
| Purchase | 1 |
| Klapaukh | 0.077 |

# Motivation

- Detecting symmetries in abstract graph is NP-Complete
- K and P assign a symmetric score to drawings of graphs, but they often disagree and the measures are not robust
(e.g., sensitive to scaling and minor perturbations)
- There are no algorithms that create symmetric graph drawings

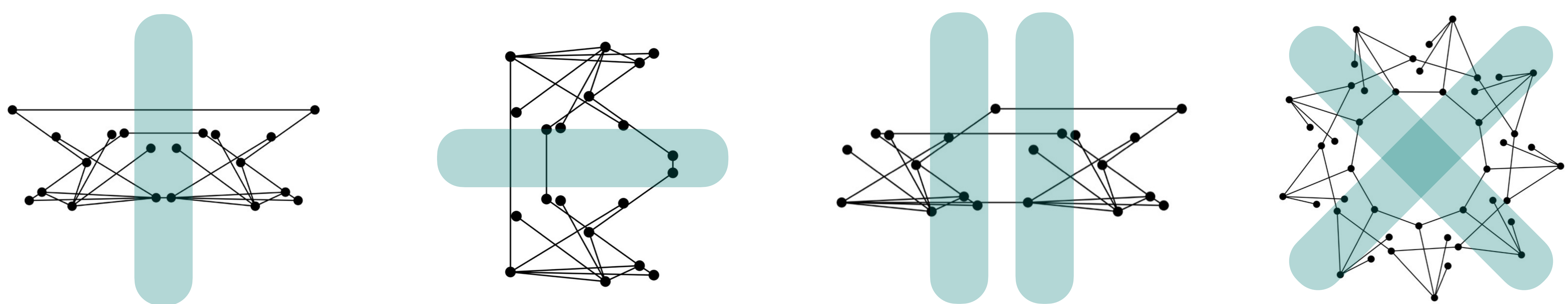


# Goals

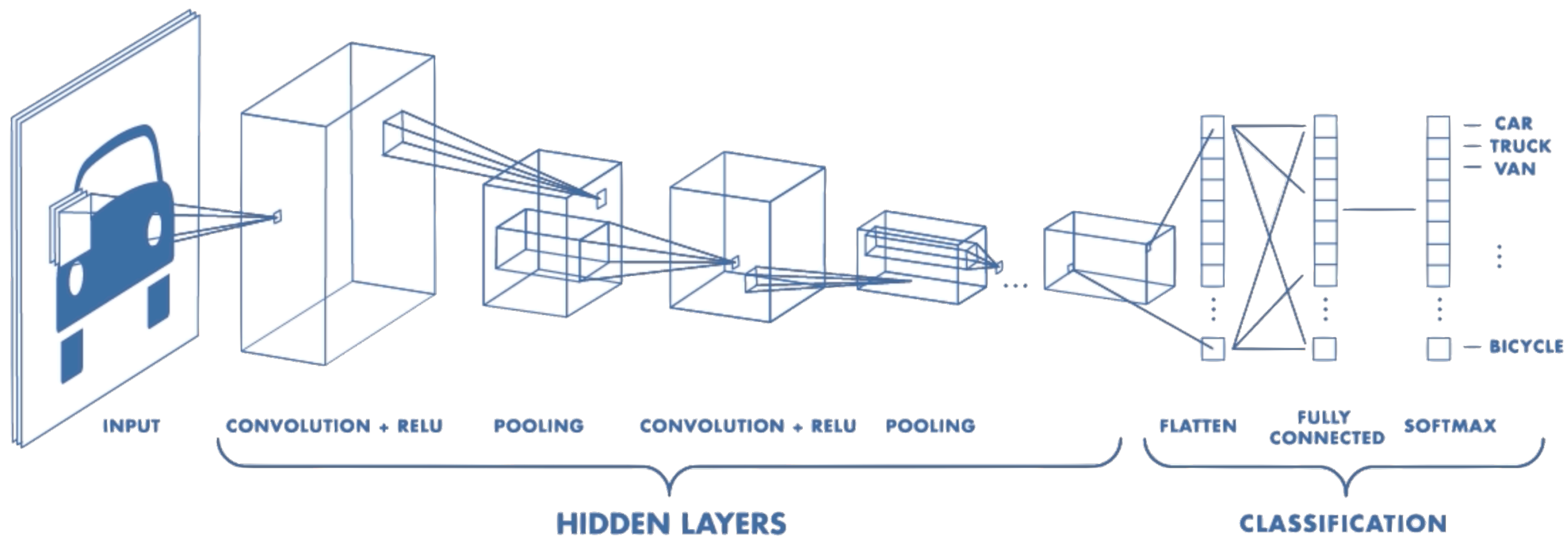
1. We would like to have a robust measure of symmetries in graph drawing

2. We would like to use such a measure to create symmetric graph drawings

3. As an initial step, we propose an machine learning approach to detect and classify symmetries in drawings of graphs



# Convolutional Neural Network: Brief introduction




- **deep learning** algorithm
- standard approach for **images classification**
- **features learning** to differentiate classes
- **weights** and biases to learn features
- **feedforward** for classification
- **backpropagation** for tuning

Keras

https://keras.io
Open source
high-level neural networks API

We train 11 Keras models from scratch using our dataset

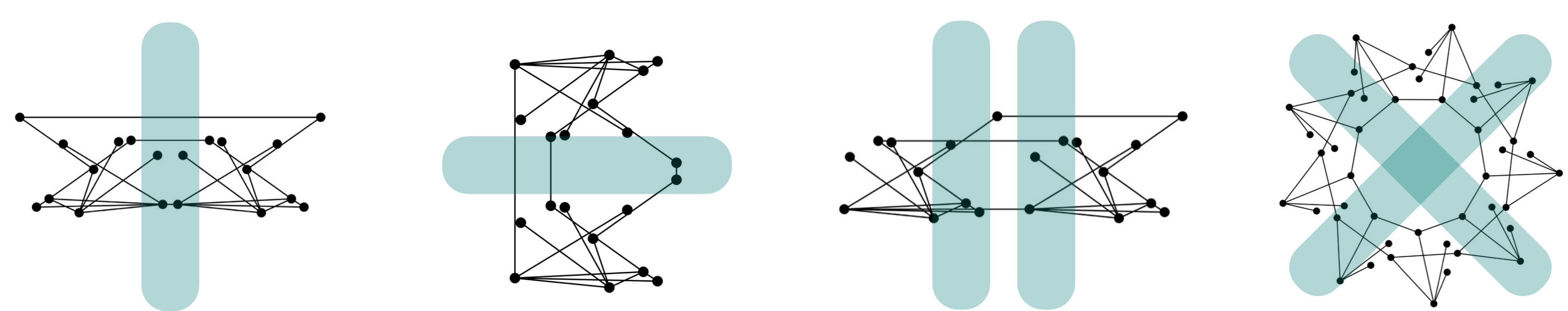
https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html

# Dataset generation

- No dataset of symmetric graph drawings suited for CNN
- No algorithm to draw symmetric graphs

We generated images for:

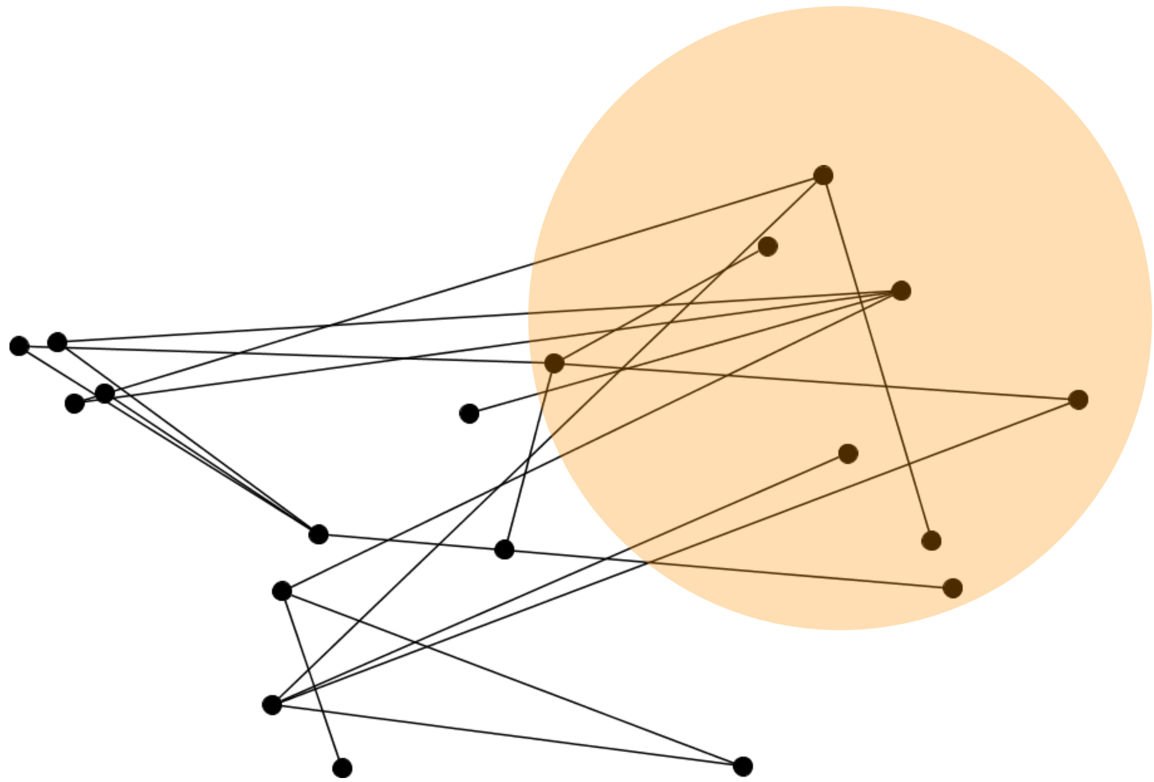
- **reflectional**: horizontal, vertical and any axis
- **rotational**
- **translational**
- **non symmetric**



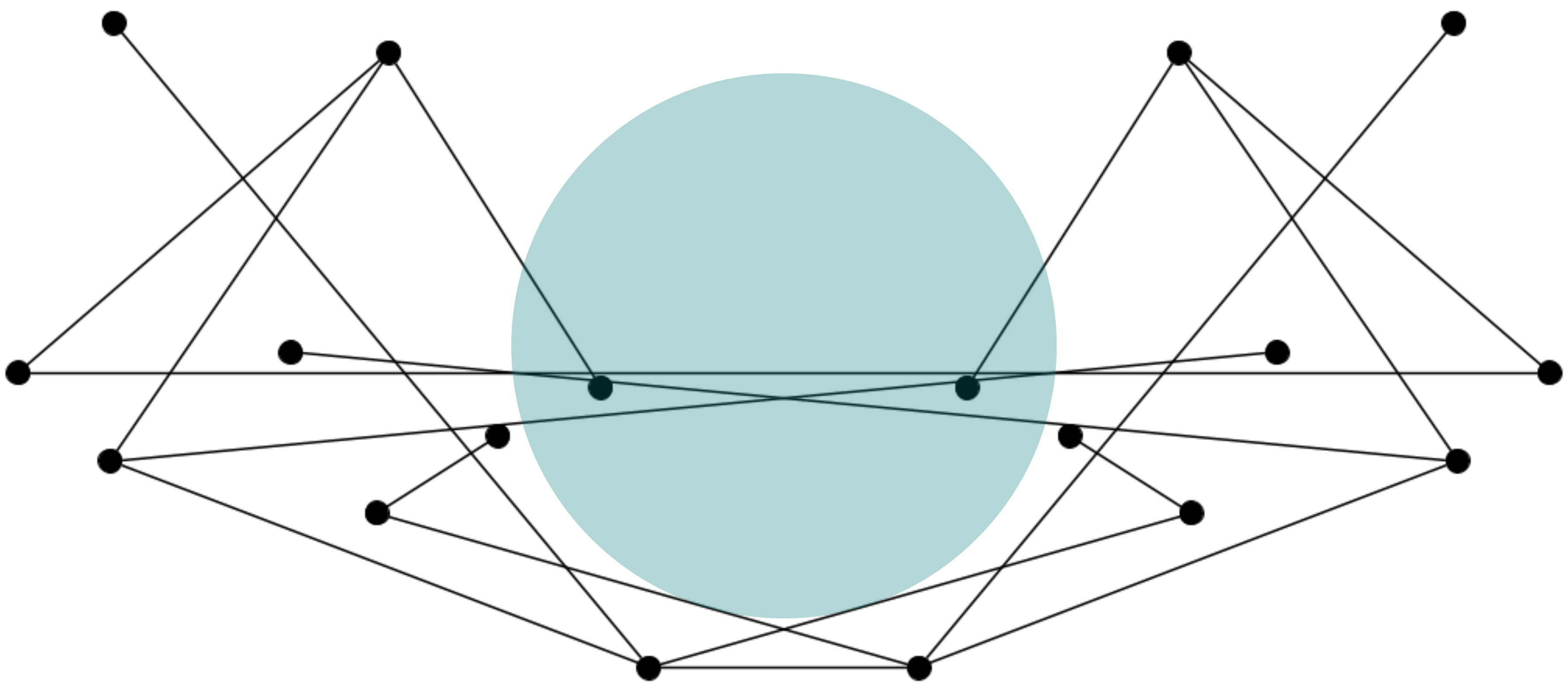
# Reflectional &

# Non symmetric generation

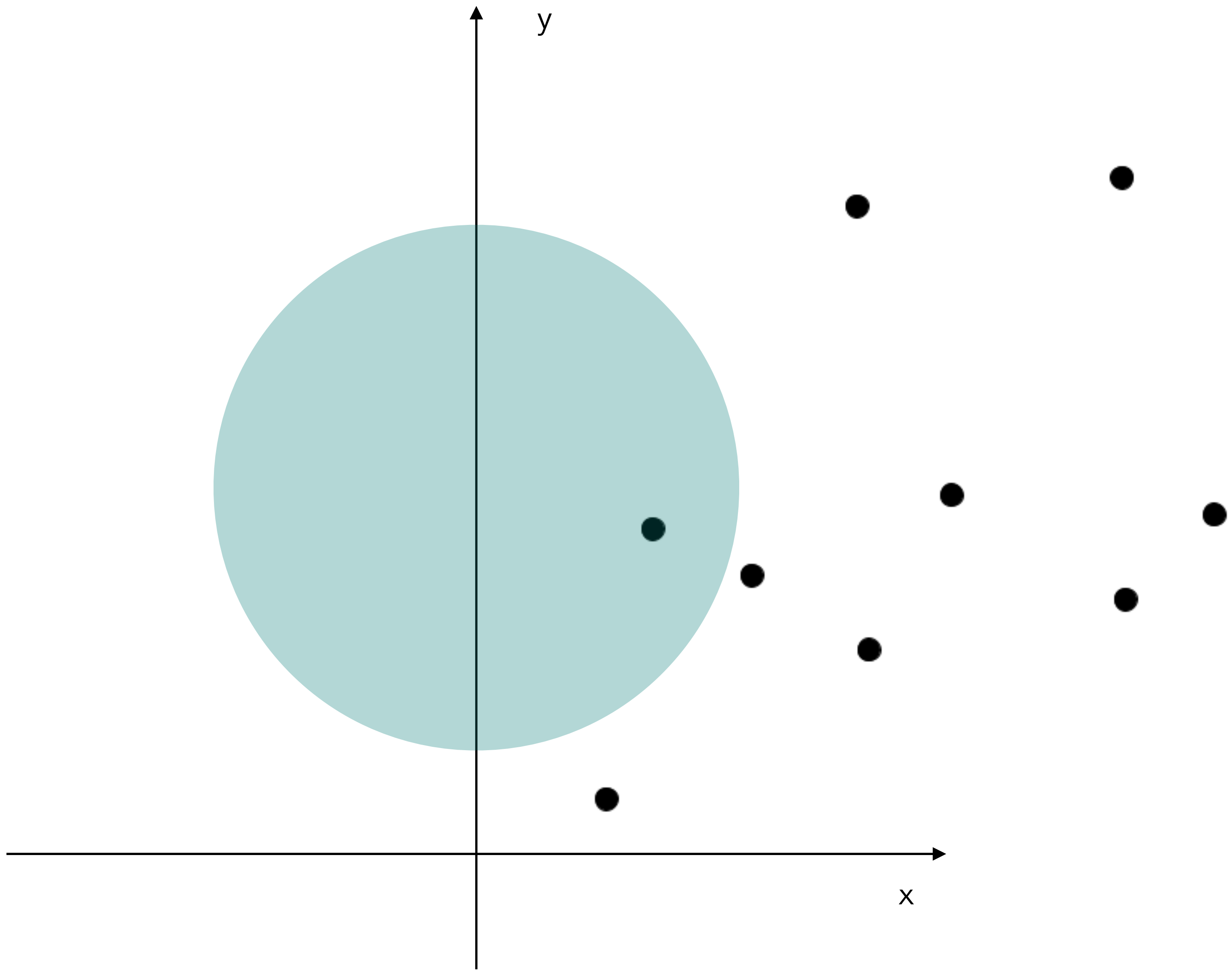
# Non symmetric layout



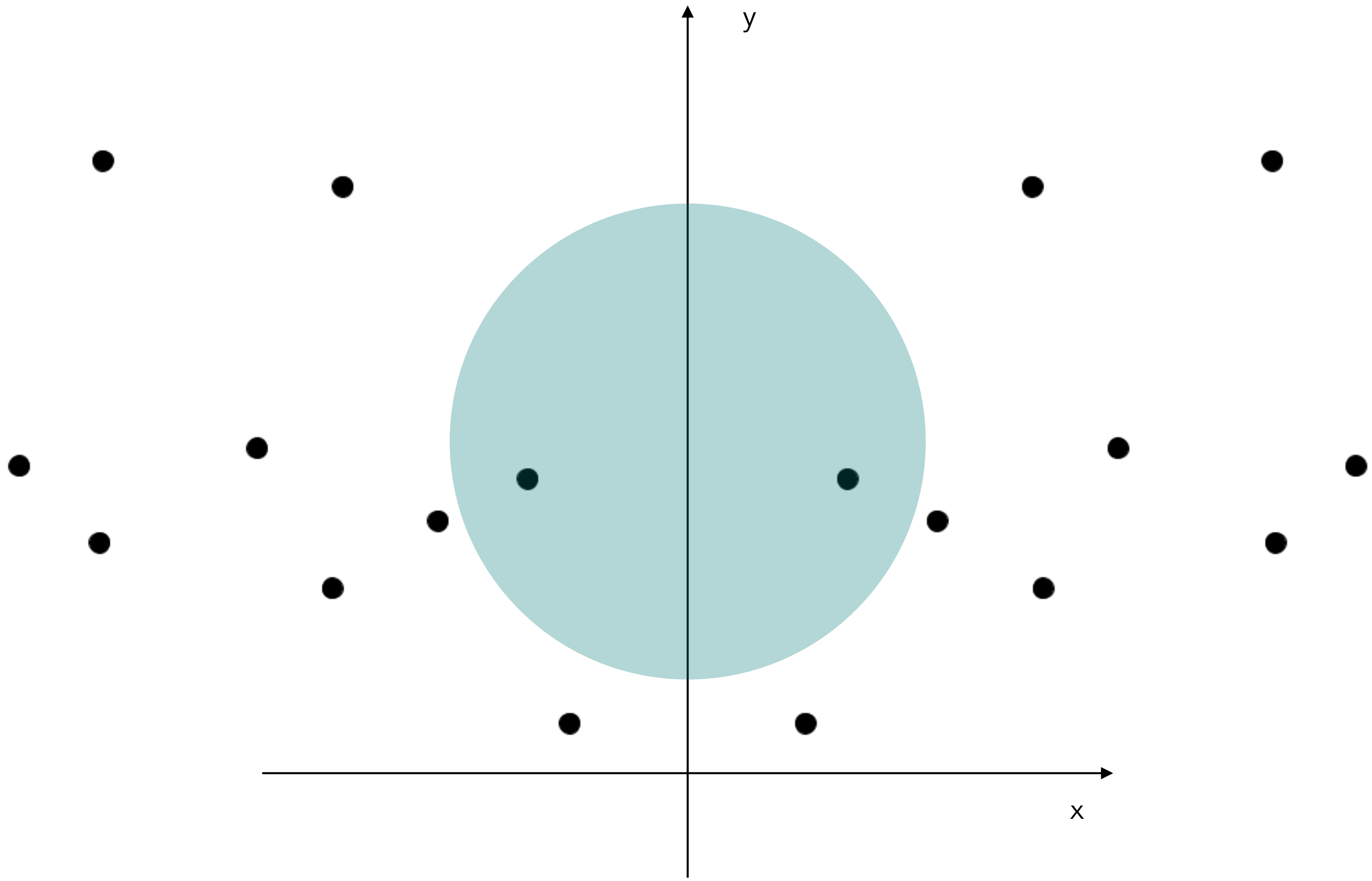
# Reflectional layouts



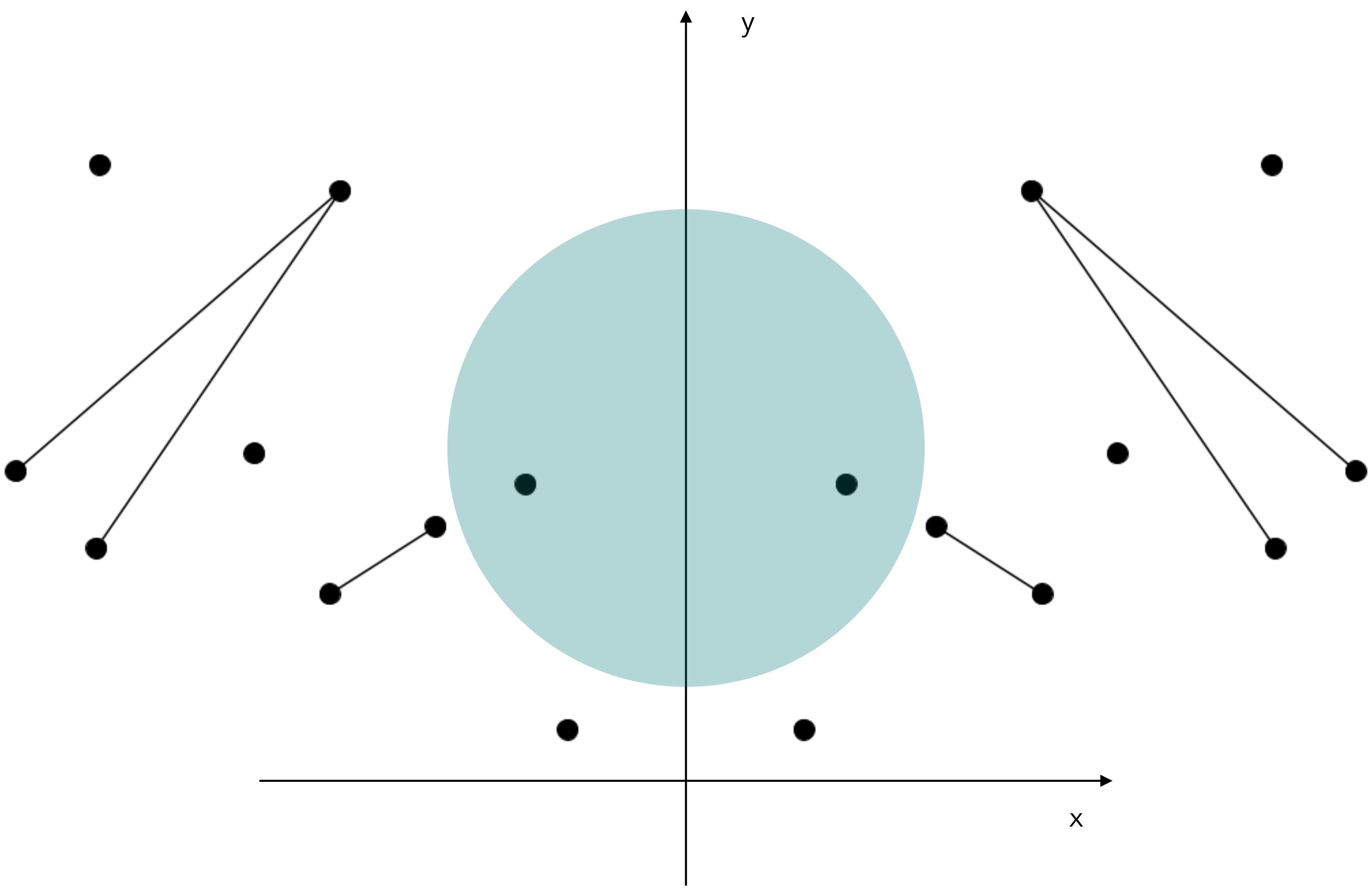
# Reflectional layouts



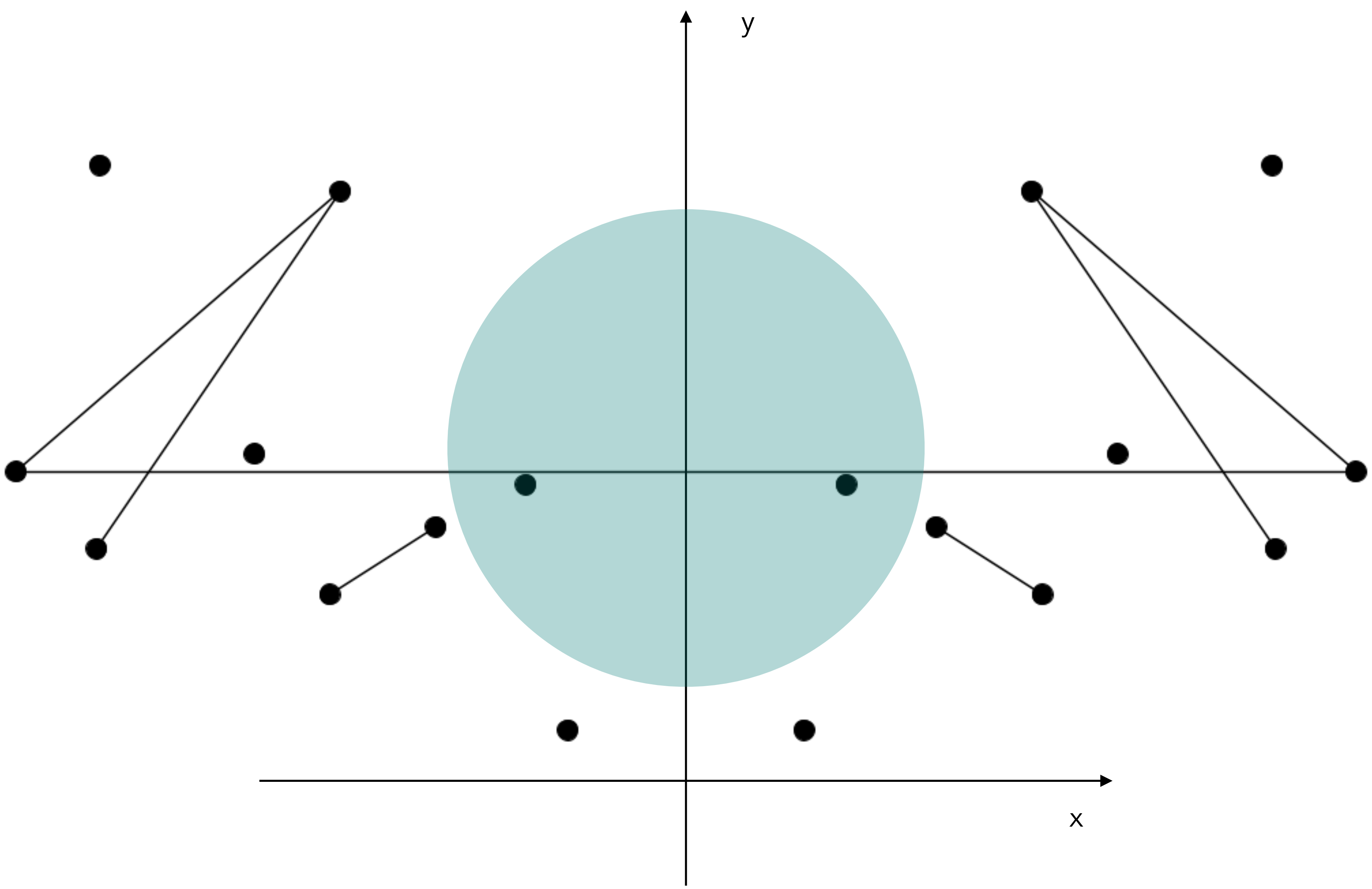
# Reflectional layouts



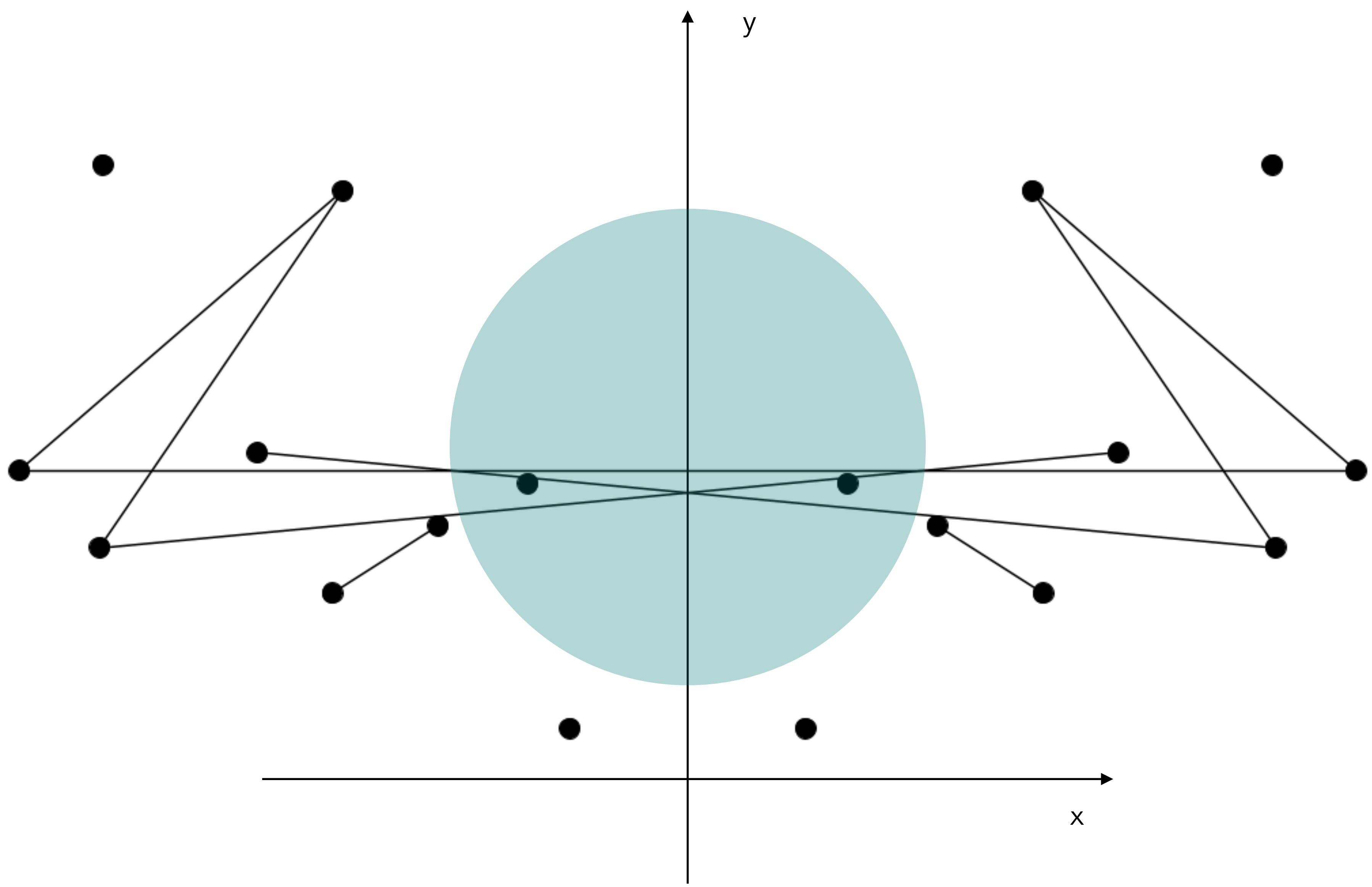
# Reflectional layouts



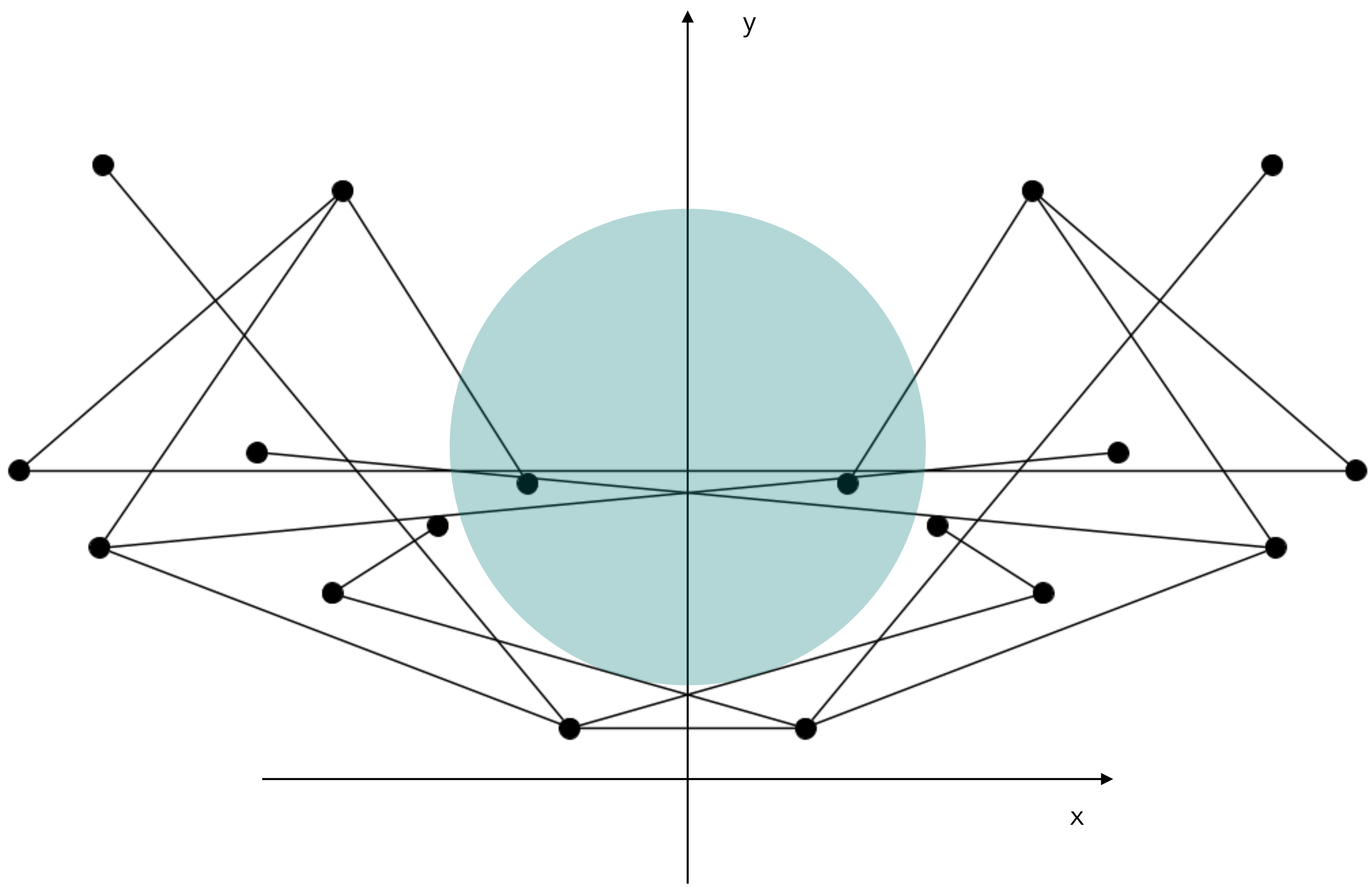
# Reflectional layouts



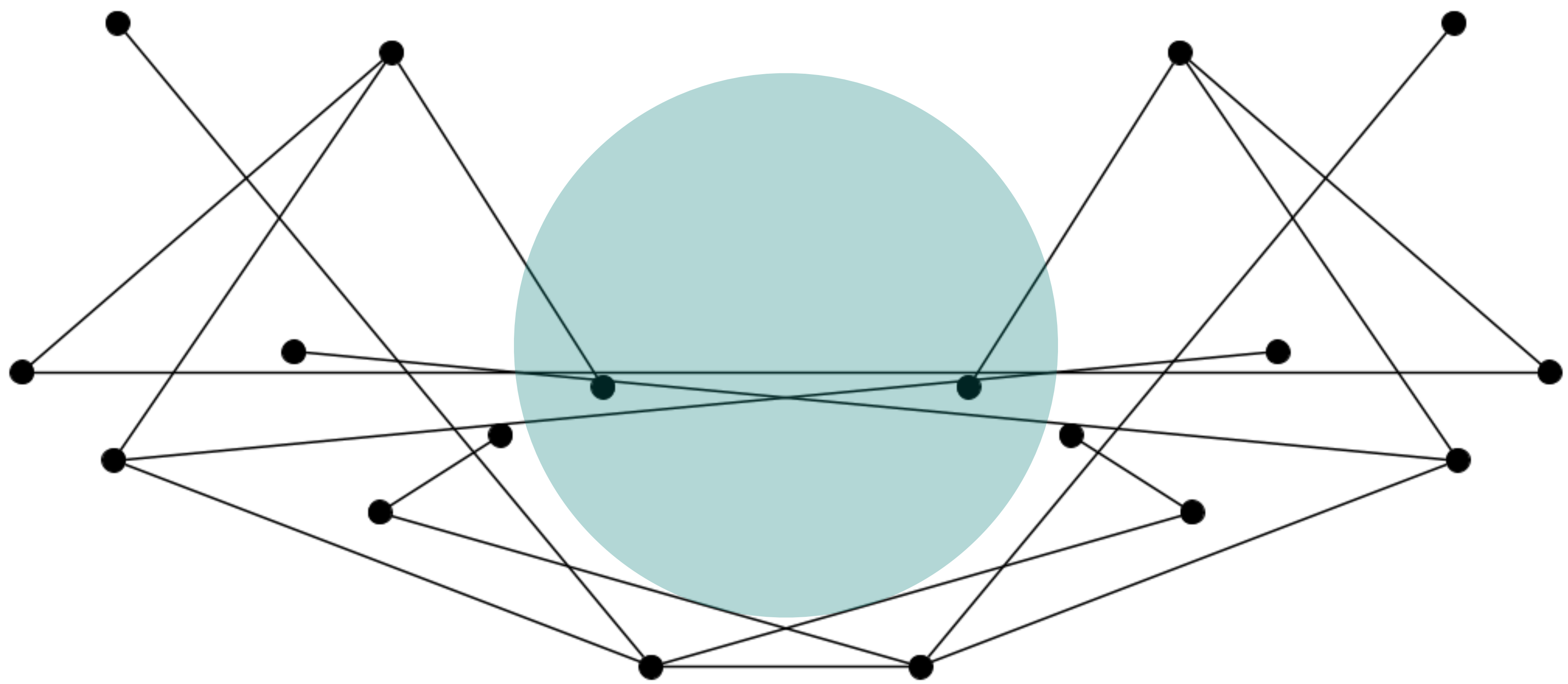
# Reflectional layouts



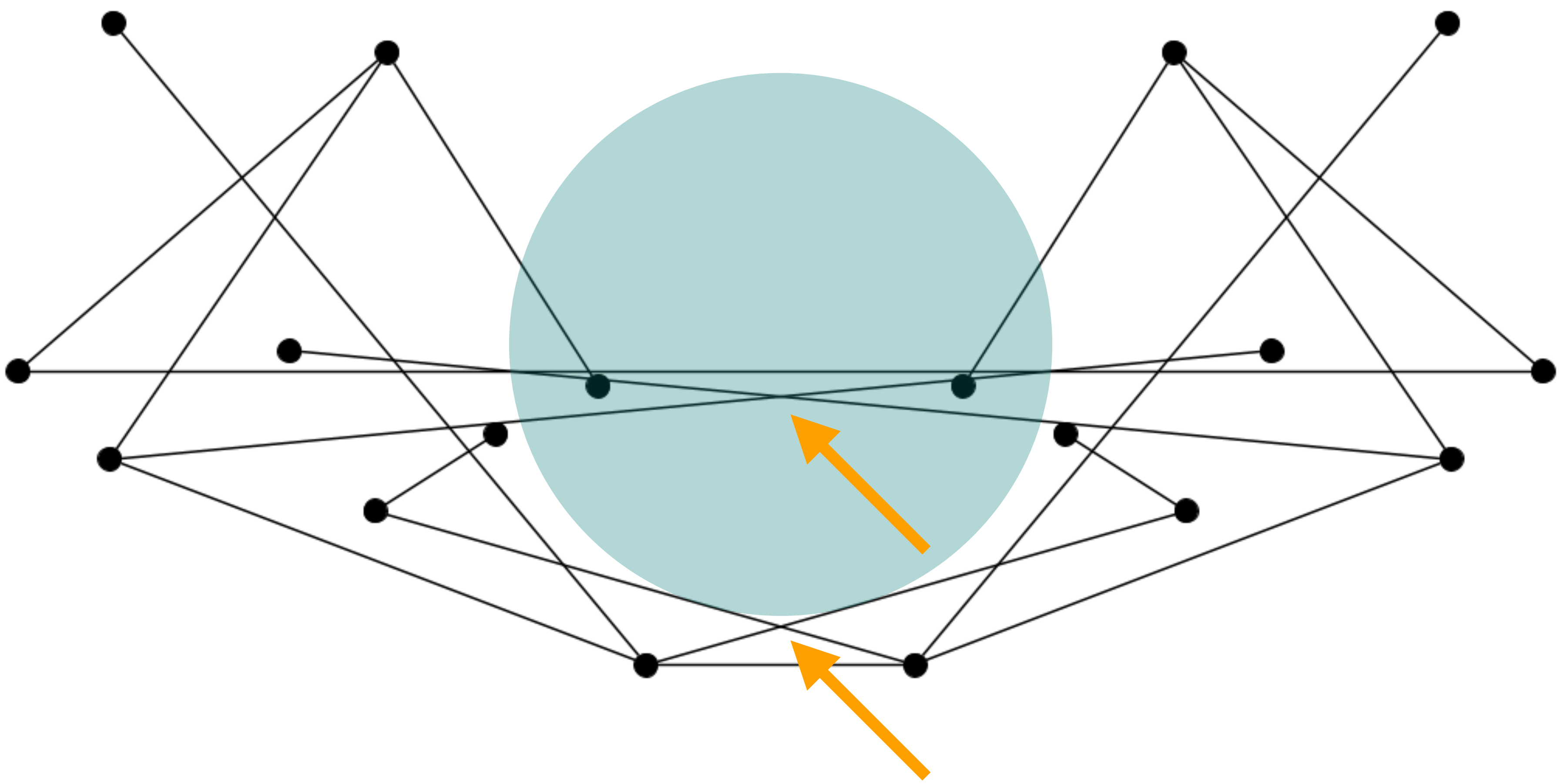
# Reflectional layouts



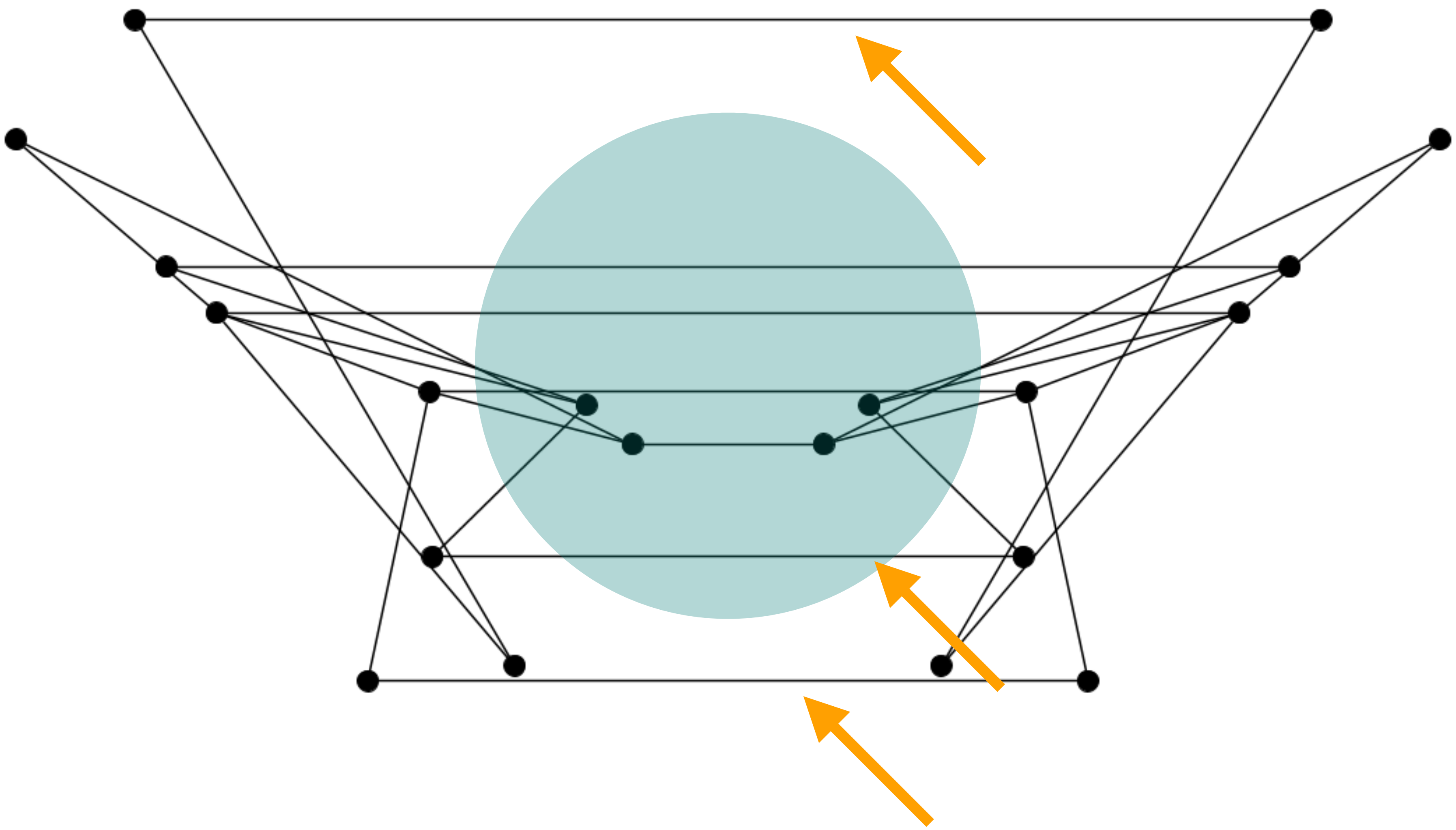
# Reflectional layouts



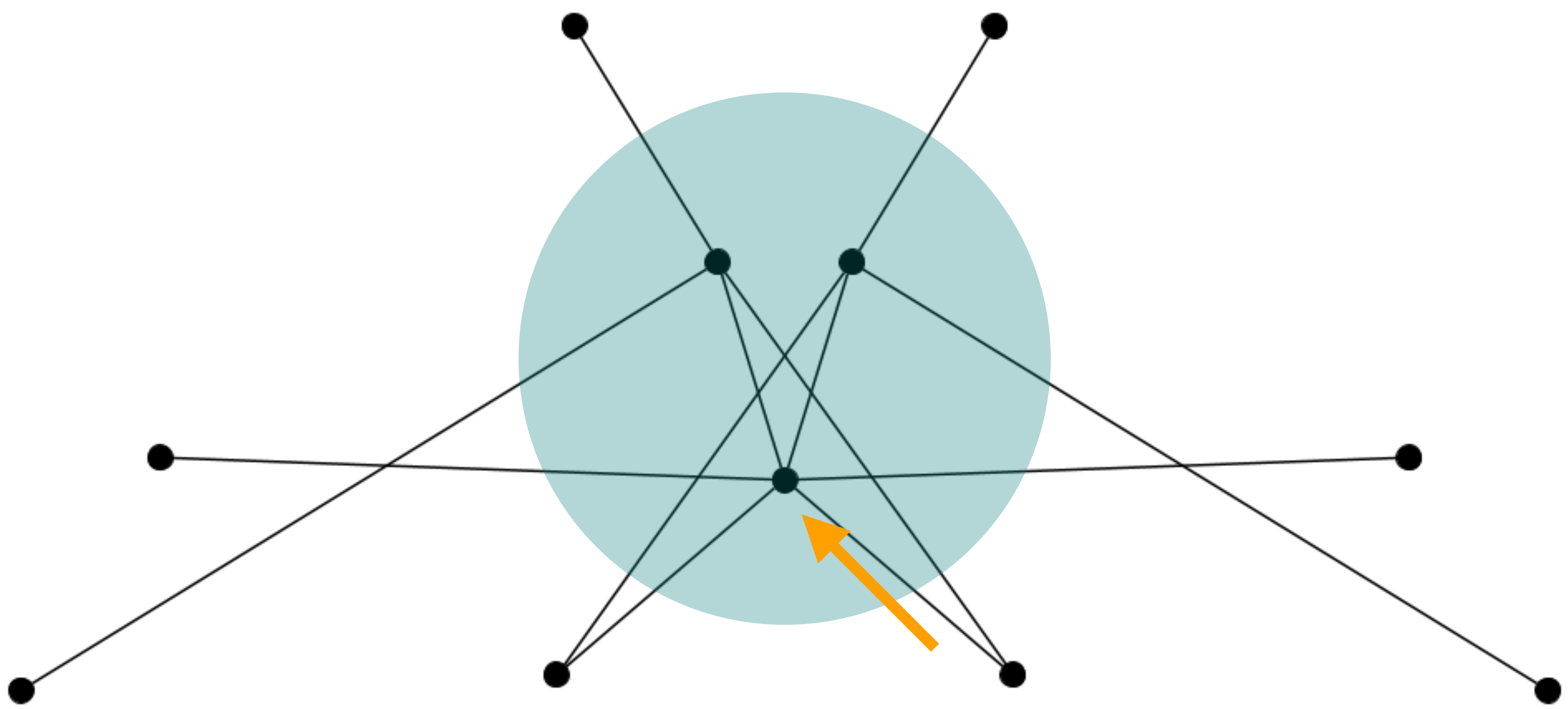
# Reflectional layout feature



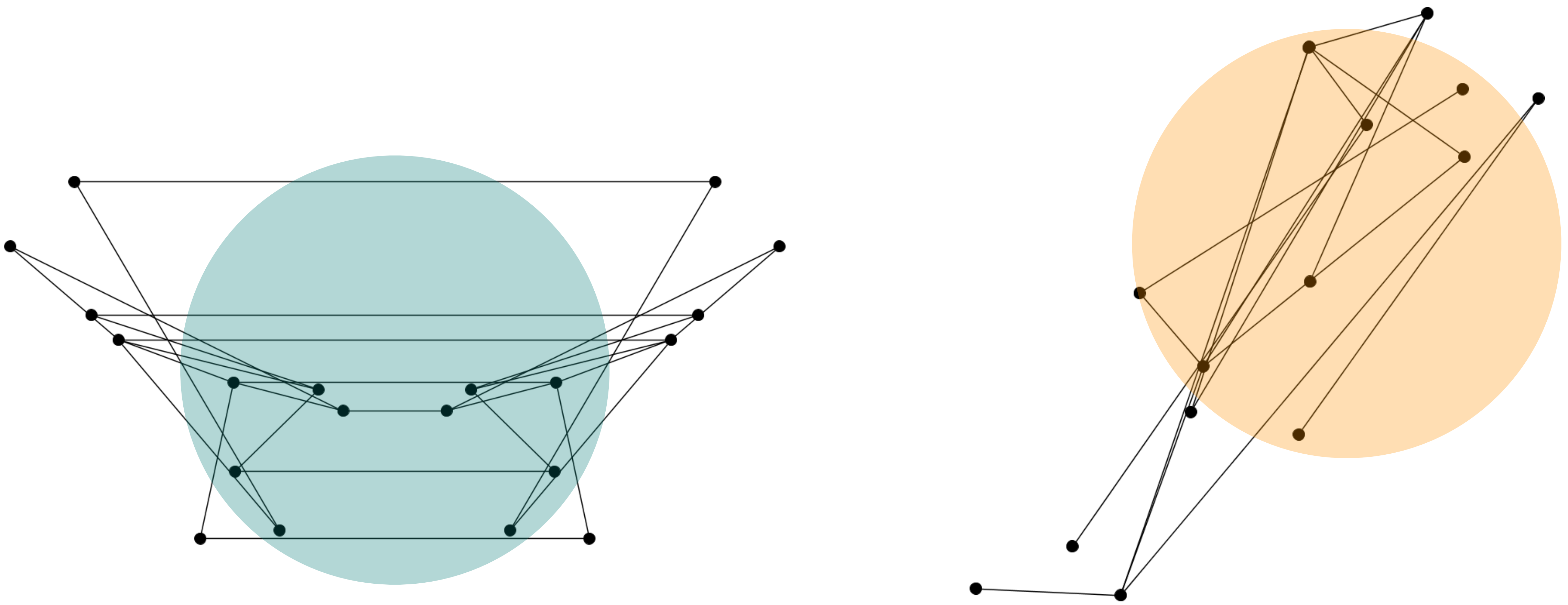
# Reflectional layout feature



# Reflectional layout feature

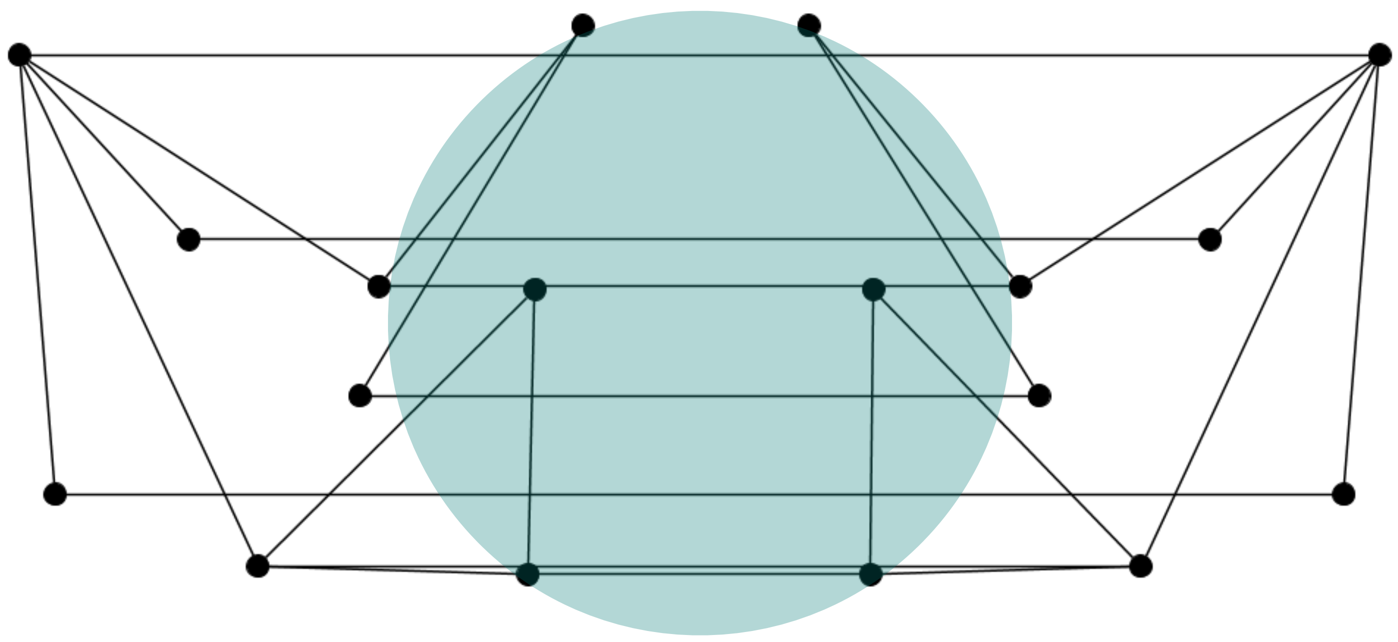


# Reflectional layout feature



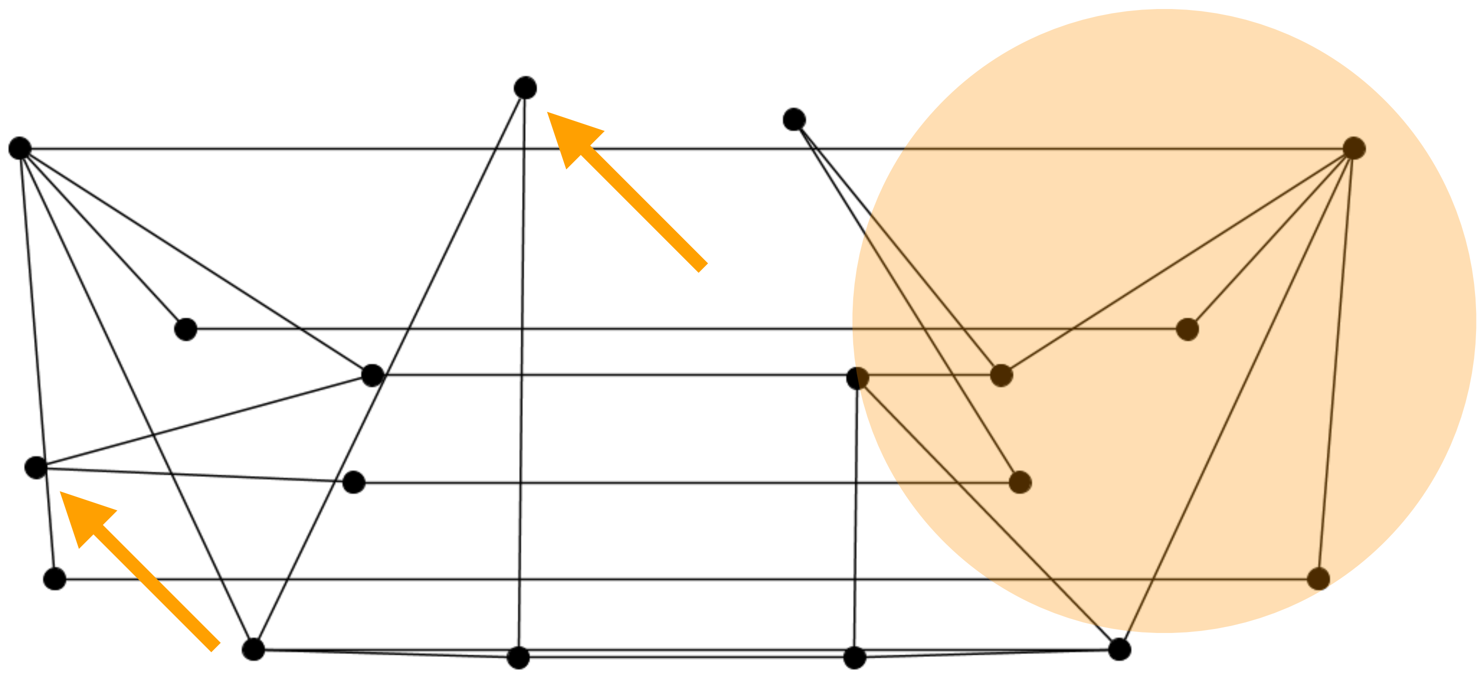
Features of a symmetric layout may not appear in a random layout

# Non symmetric pseudo random layout



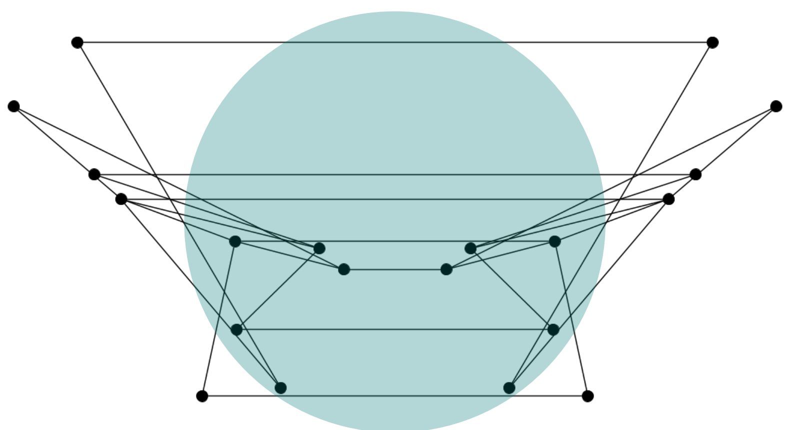
Non symmetric layout generated starting from a symmetric layout and with a little perturbation of few vertices

# Non symmetric pseudo random layout

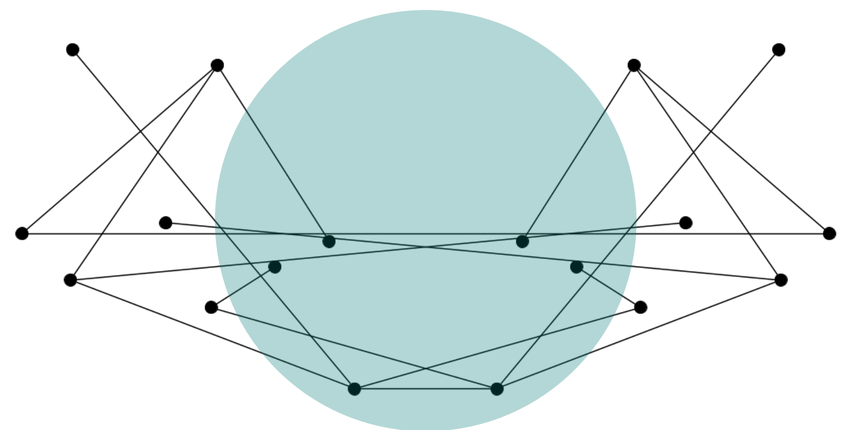


Non symmetric layout generated starting from a symmetric layout and with a little perturbation of few vertices

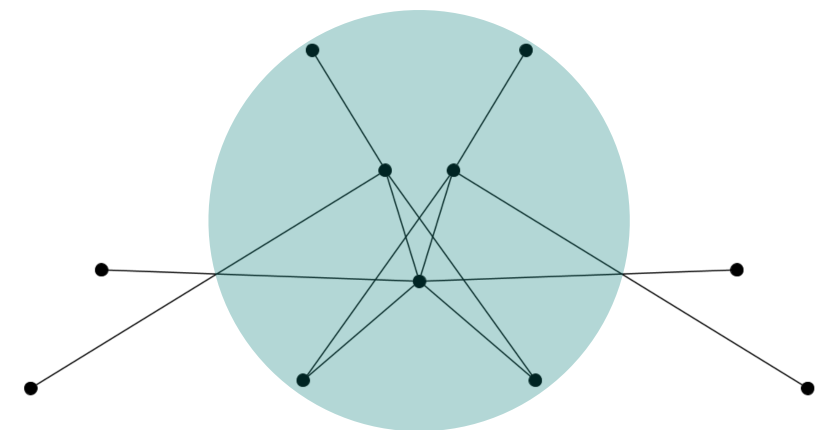
# Layouts



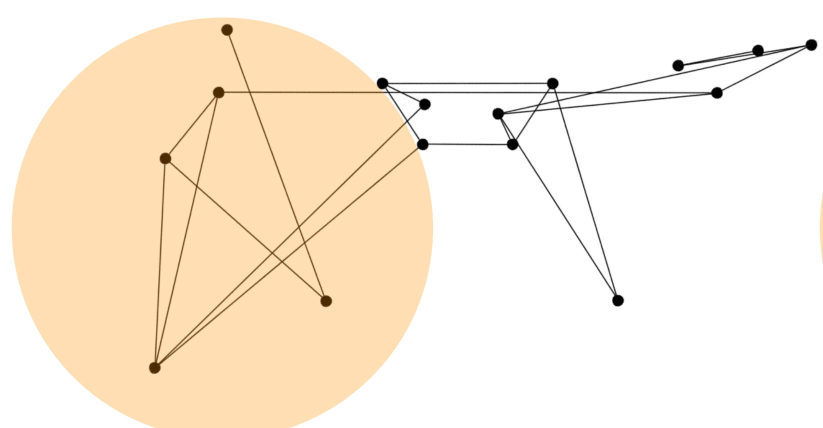
Parallel



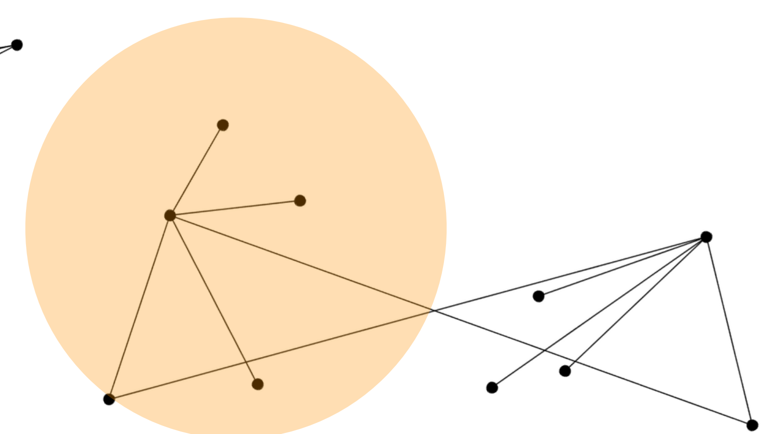
Crossings



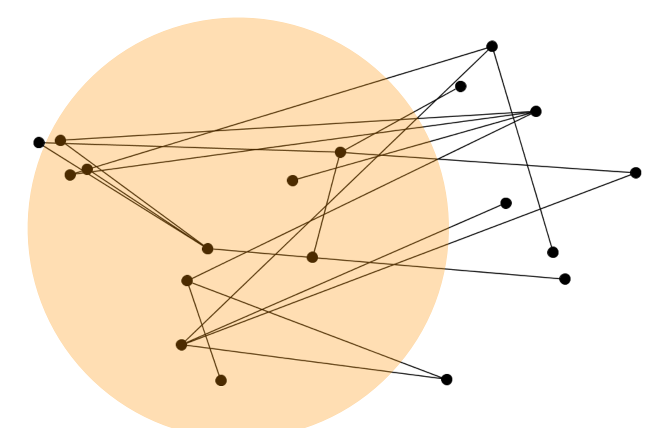
Odd |V|



Pseudorandom
parallel

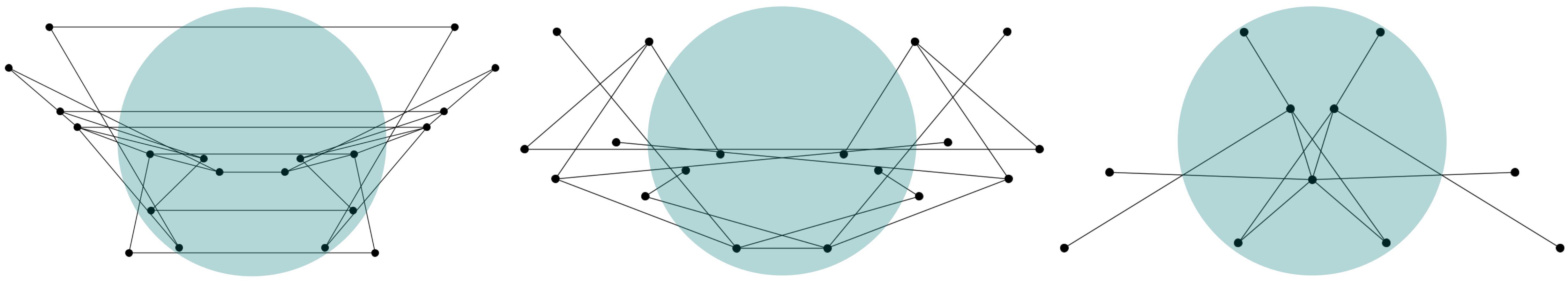


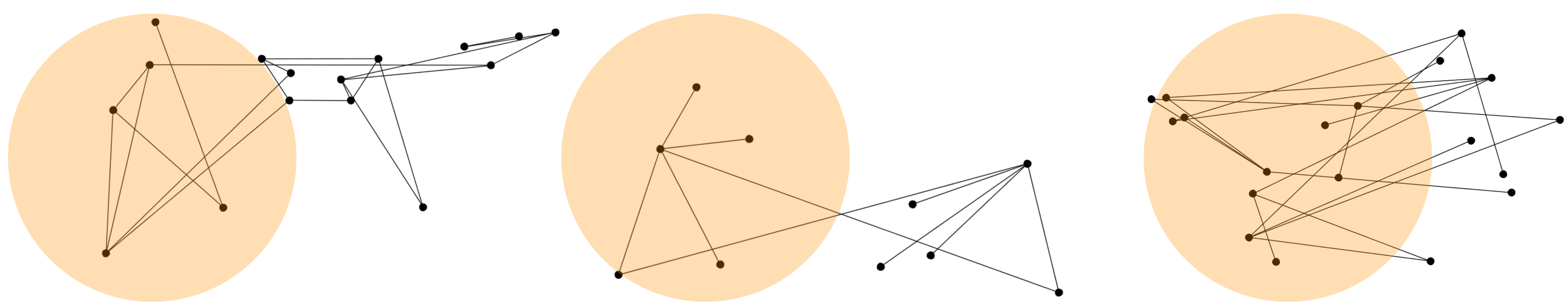
Pseudorandom
crossings



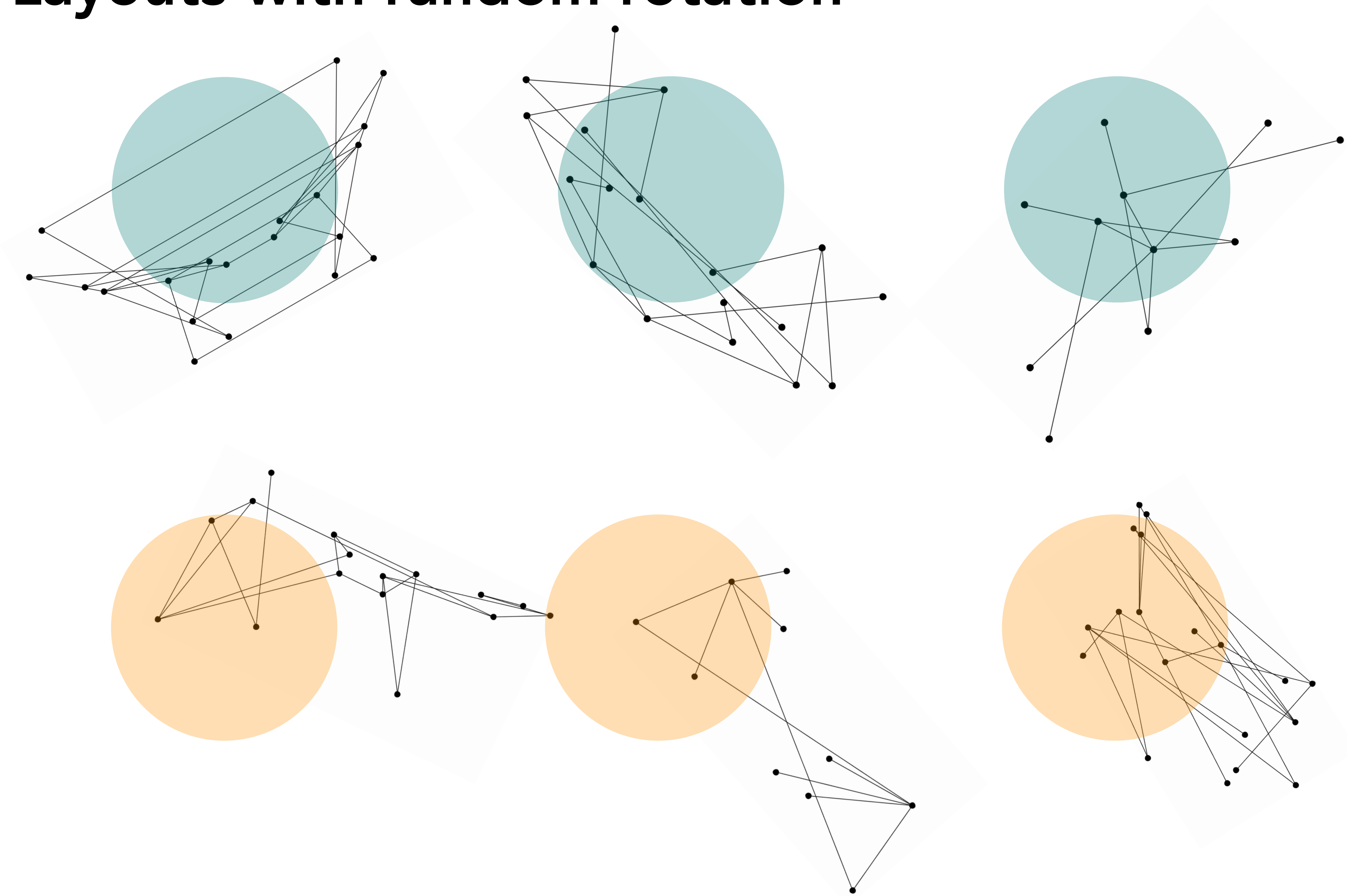
Random

# Layouts





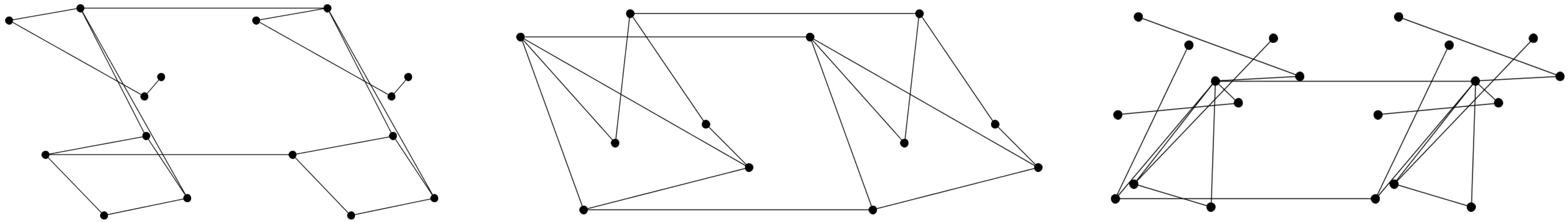
# Layouts with random rotation



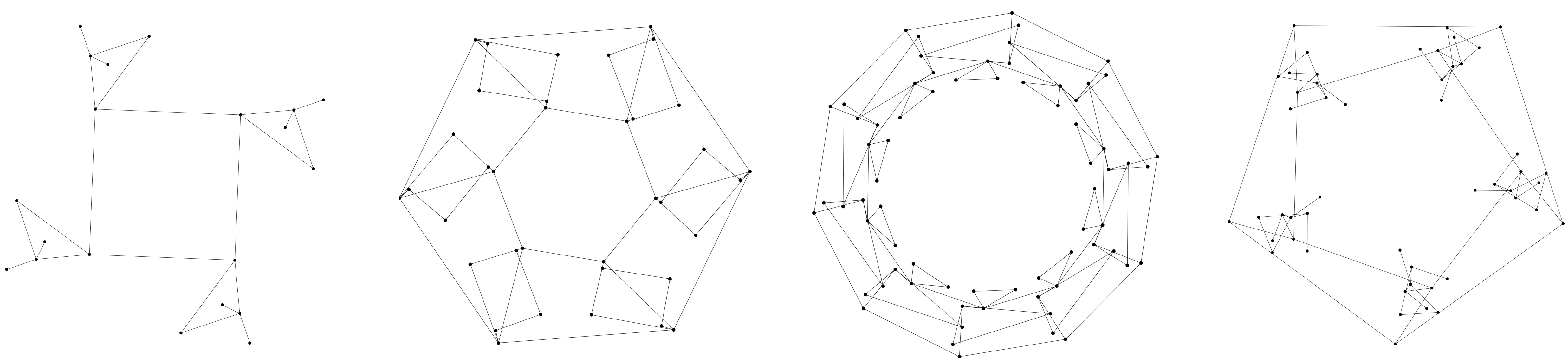
# Translational &

# Rotational generation

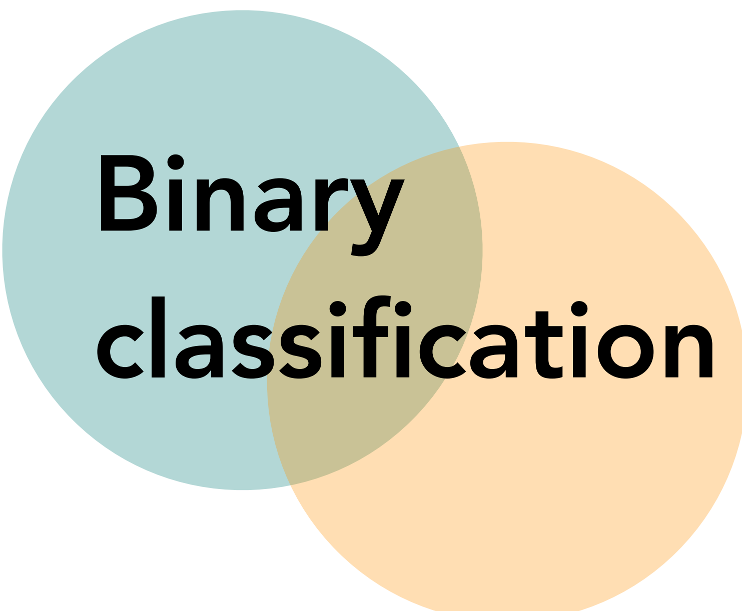
## Translational



## Rotational



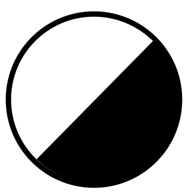
# Experiments



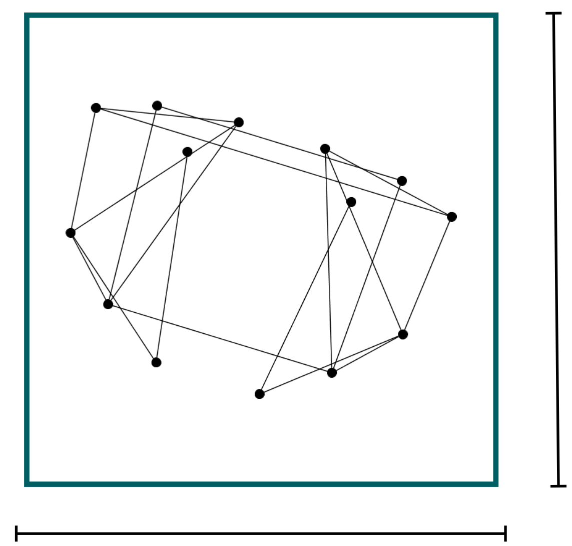
Binary classification

Multi-class classification

# Image setup

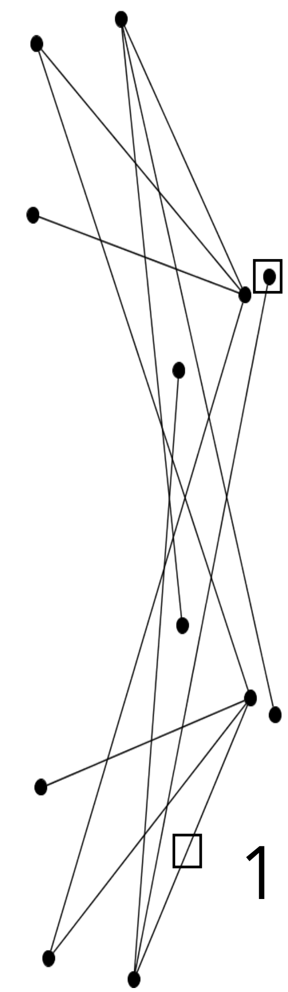


B/W Images



200 px

200 px



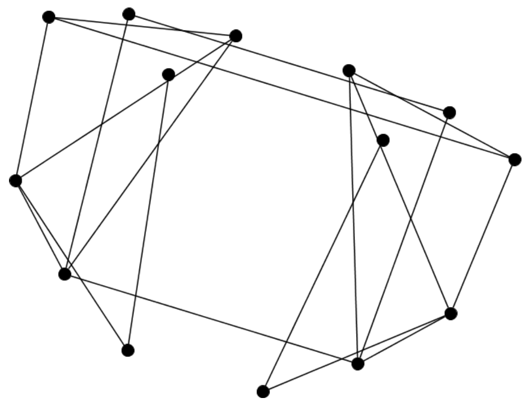
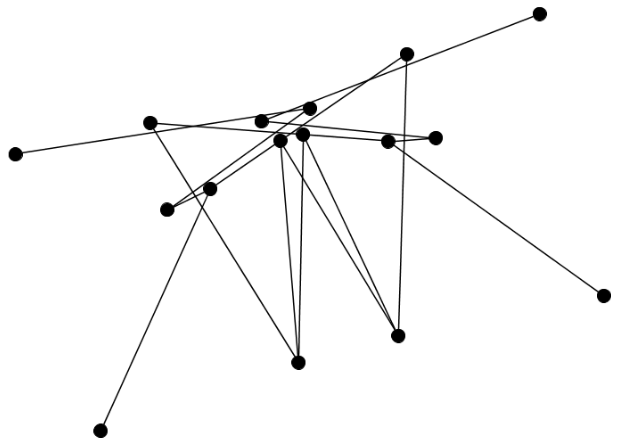
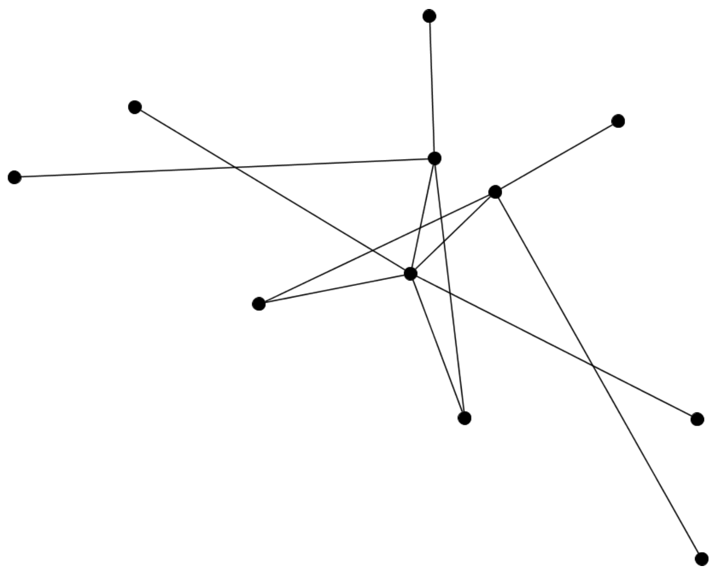
# Binary Classification:
# Is this layout symmetric?

Yes

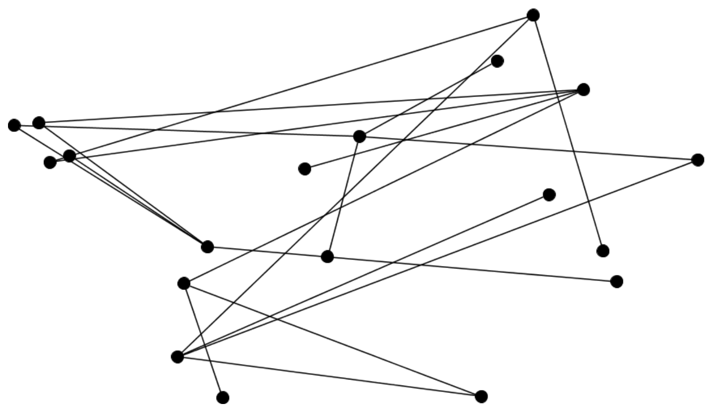
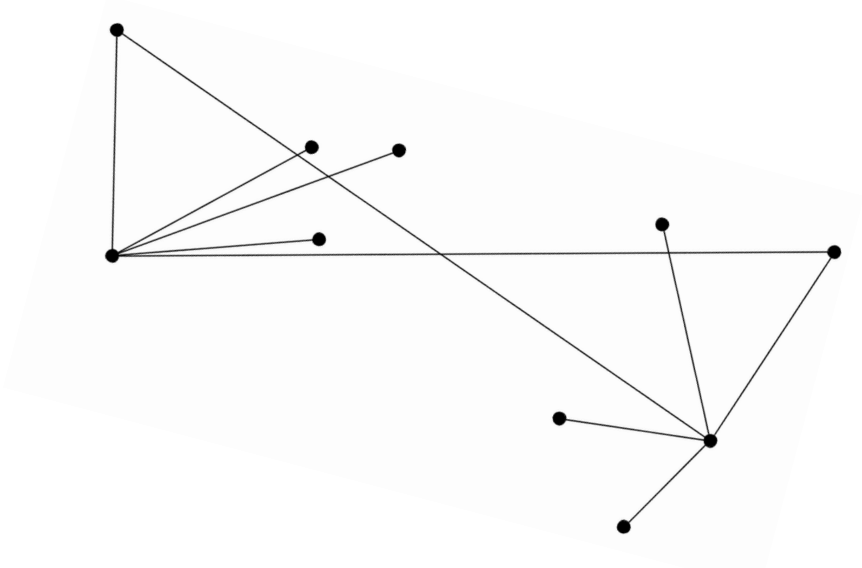
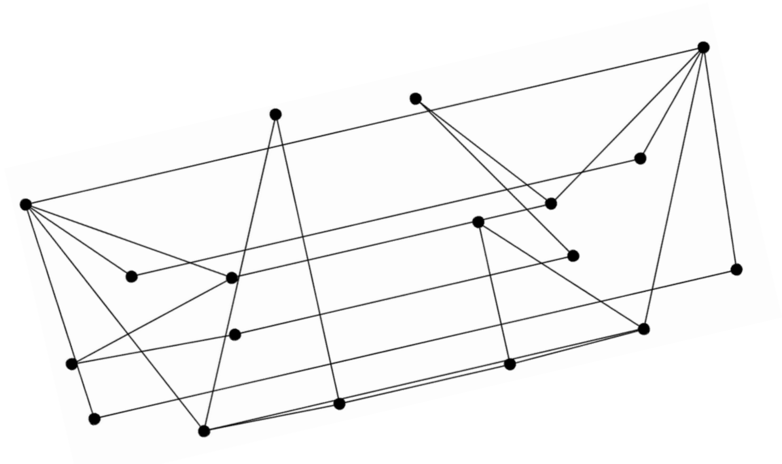
No

# Types of layout

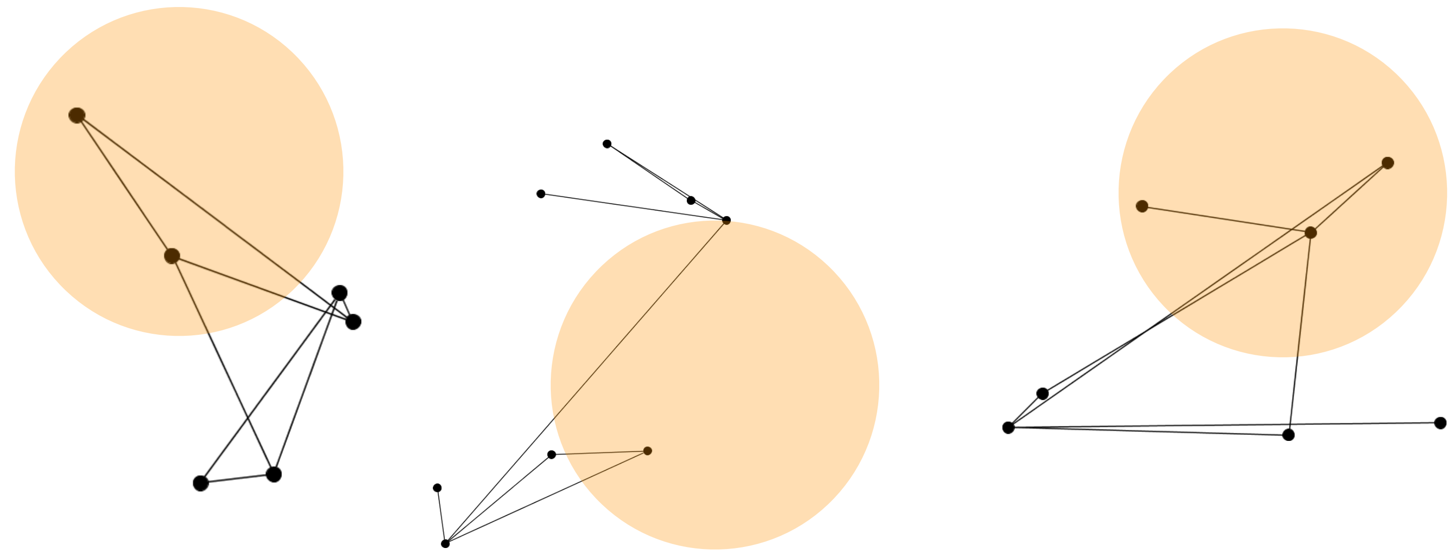
## Symmetric

## Non symmetric

# Types of layout

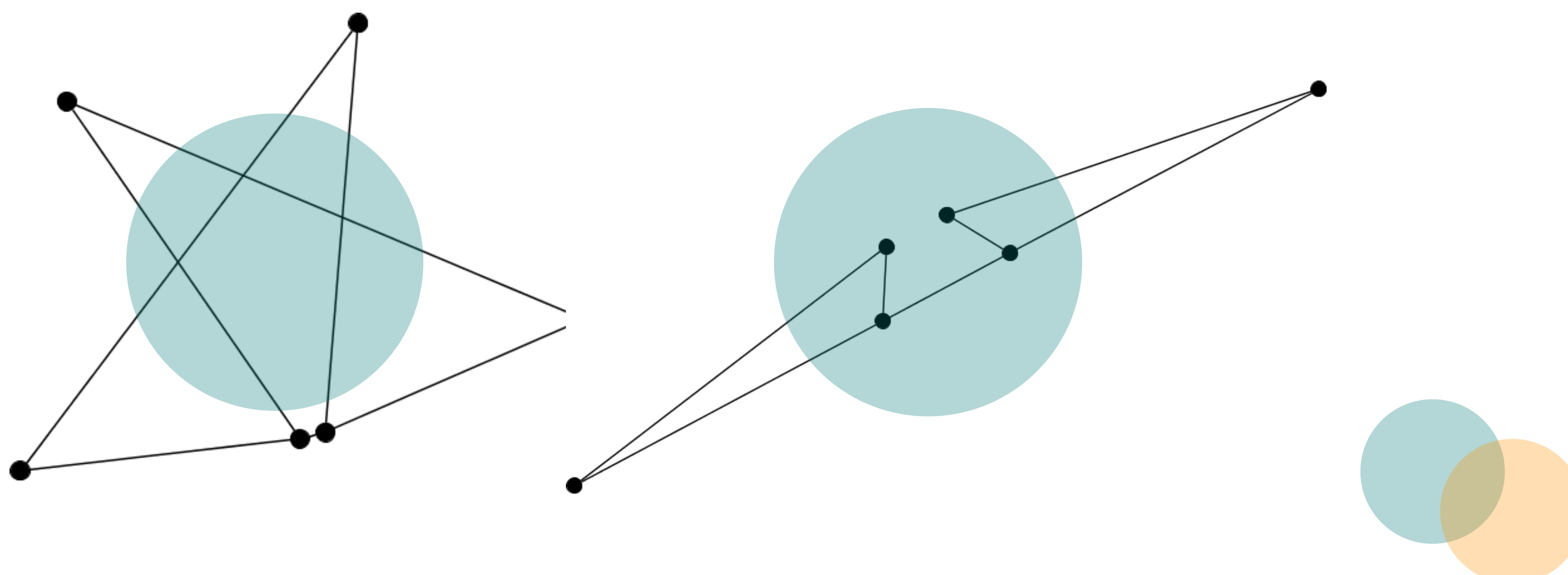


Vertices ∈ [4 , 8]

Edges ∈ [|V|, 1.2 * |V|]

50%

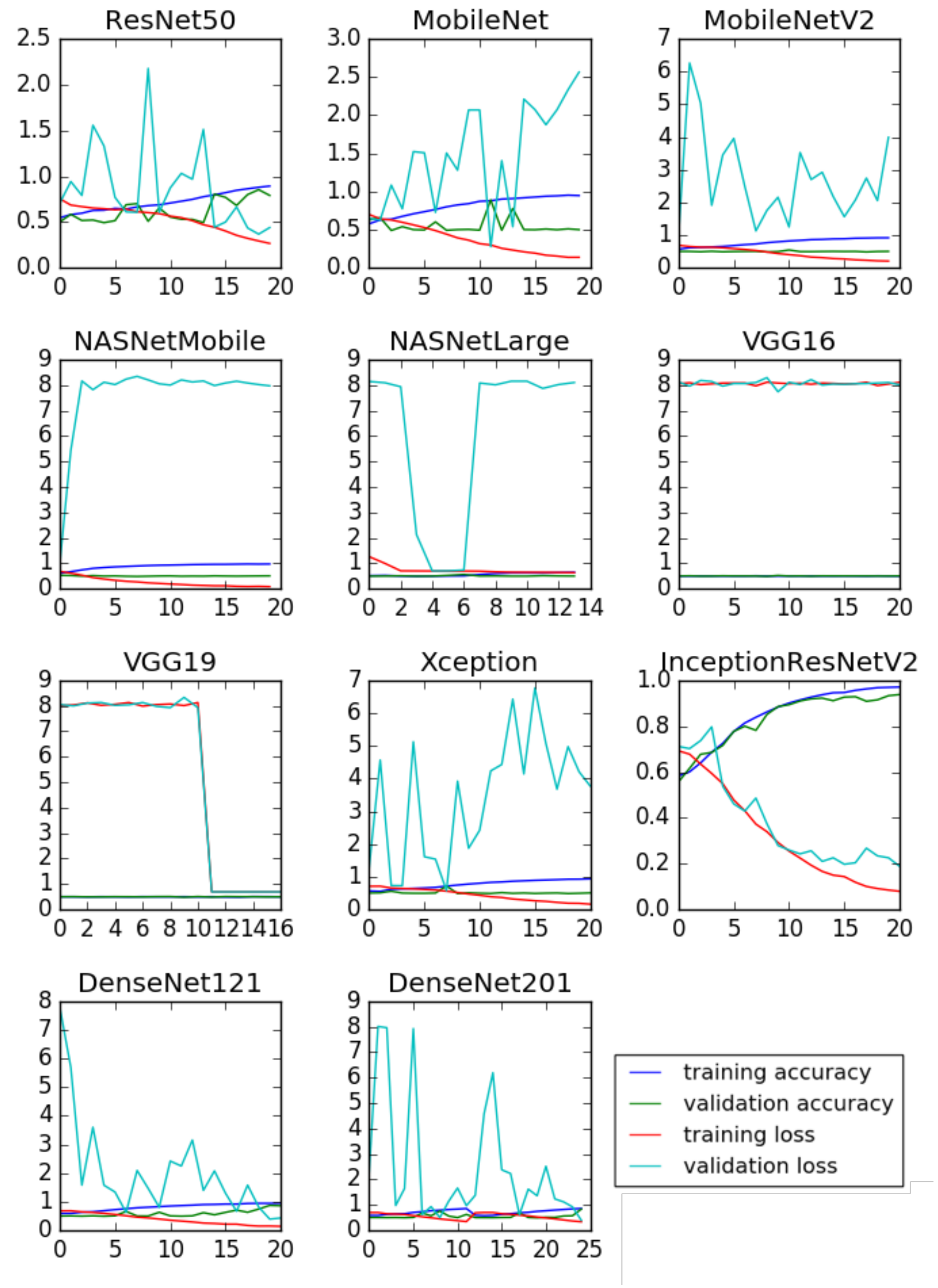
50%



# Experimental setup

- Training Images: 12k
- Validation Images: 2k
- Testing Images: 2k

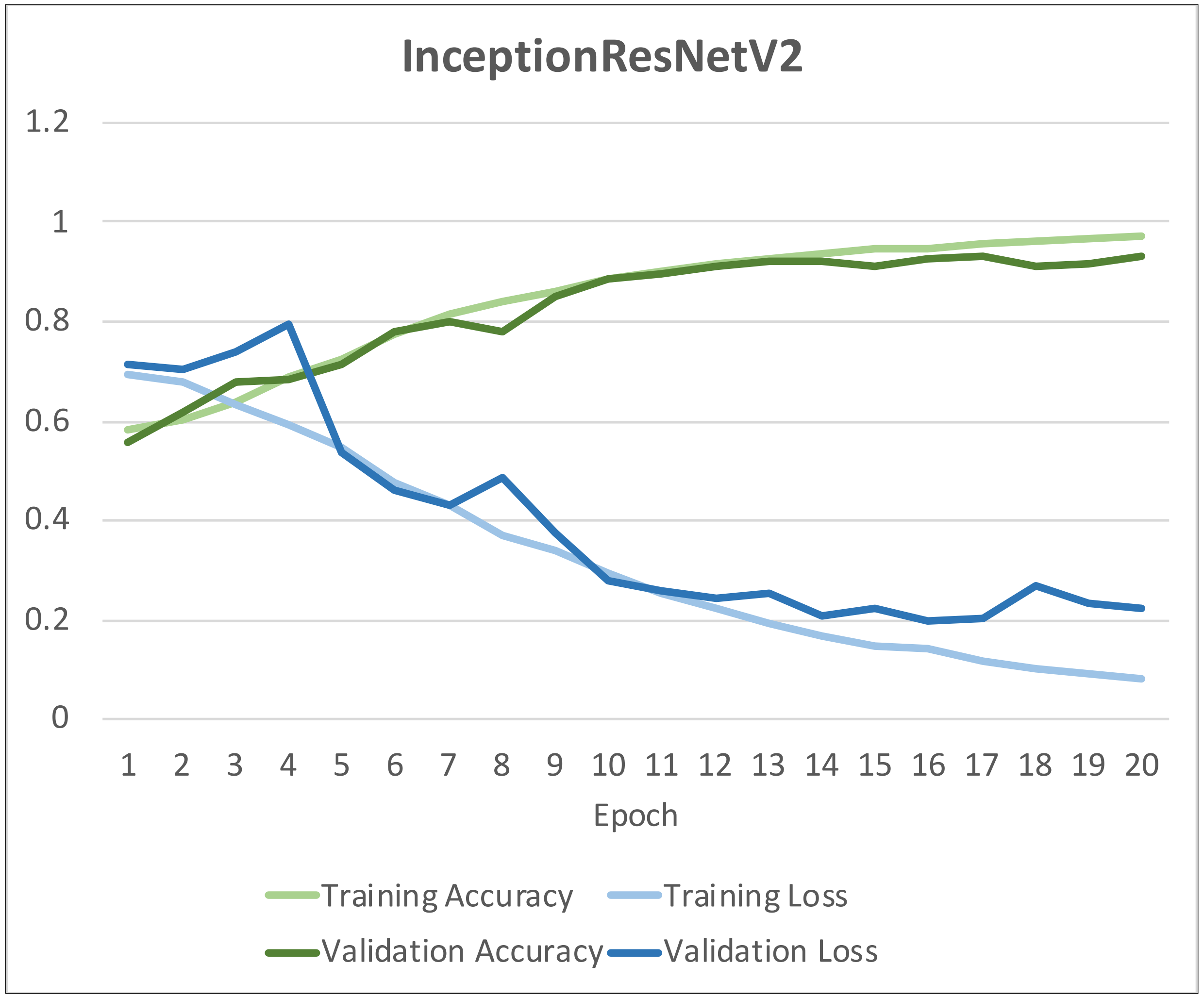
- Batch size: 16
- Epochs: 20



# Results

- Training Images: 12k
- Validation Images: 2k
- Testing Images: 2k

- Batch size: 16
- Epochs: 20




Best model:

- **InceptionResNetV2**

**Accuracy: 92%**

# Cross validation: Purchase and Klapaukh

P and K are not designed for classification, but do provide a symmetry score in the range [0, 1]

We use a 0.5 threshold to classify the scored instances as follows:

- symmetric, if the score is >0.5
- non-symmetric, otherwise

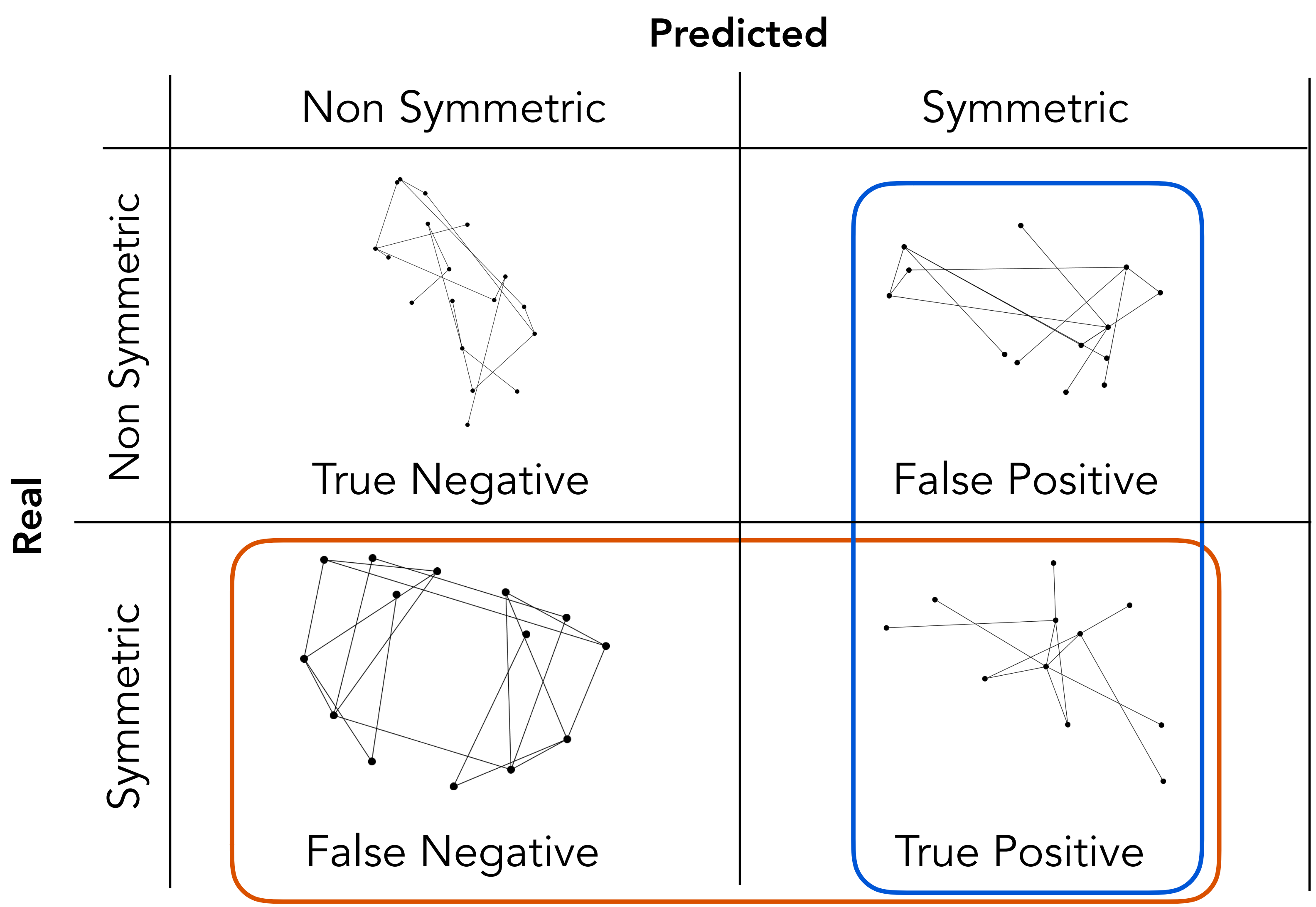
We use the same 2000 images

# Cross validation: Purchase and Klapaukh

**Recall**:
How many predicted elements are relevant (symmetric)?
TP/(FN+TP)

**Precision**:
How many relevant (symmetric) items are selected?
TP/(FP+TP)

**F1-Score**:
Harmonic mean of precision and recall
2*R*P/(R+P)



# Cross validation: Purchase and Klapaukh

**Recall**:
How many predicted elements are relevant (symmetric)?
TP/(FN+TP)

**Precision**:
How many relevant (symmetric) items are selected?
TP/(FP+TP)

**F1-Score**:
Harmonic mean of precision and recall
2*R*P/(R+P)

| | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| **Purchase** | 82% | 0.67 | **0.96** | 0.79 |
| **Klapaukh** | 82% | 0.80 | 0.86 | 0.83 |
| **InceptionResNet** | **92%** | **0.90** | 0.95 | **0.93** |

# Multi-class Classification:
# What type of symmetry does it show?

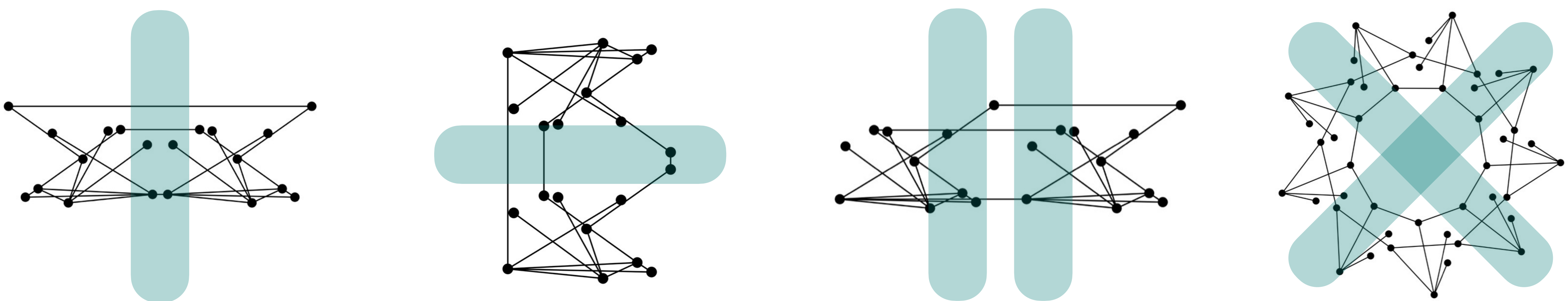
Vertical

Horizontal

Translational

Rotational

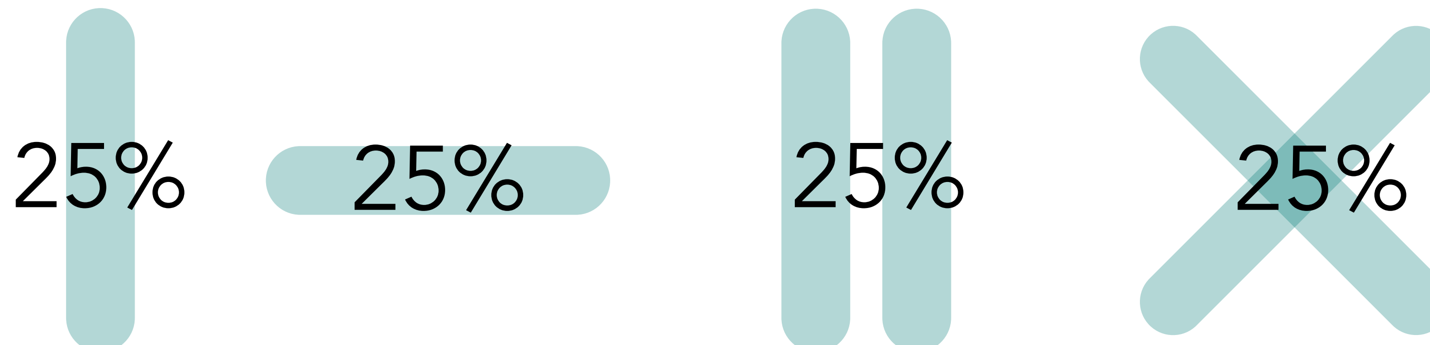
# Types of layout

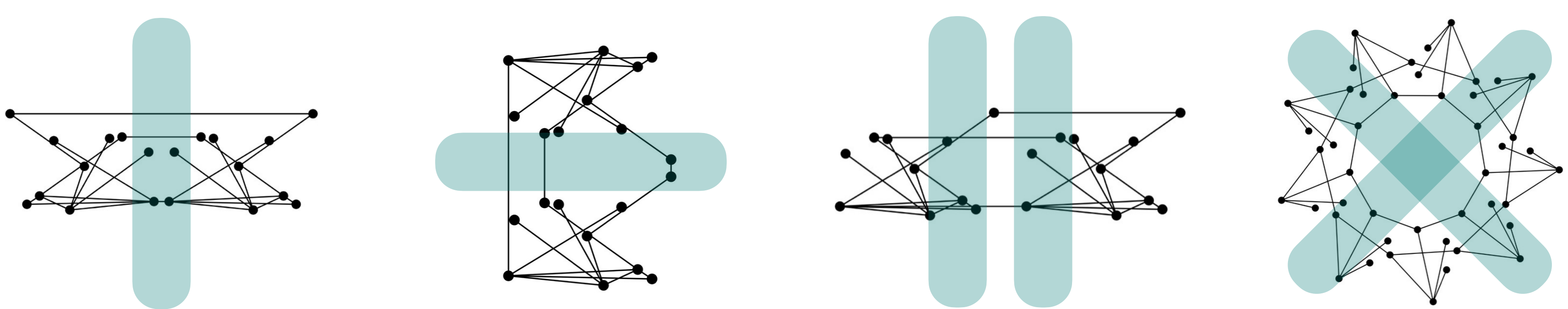


# Experimental setup

Vertices $\in [10\ ,\ 20]$

Edges $\in [|V|,\ 1.2 * |V|]$

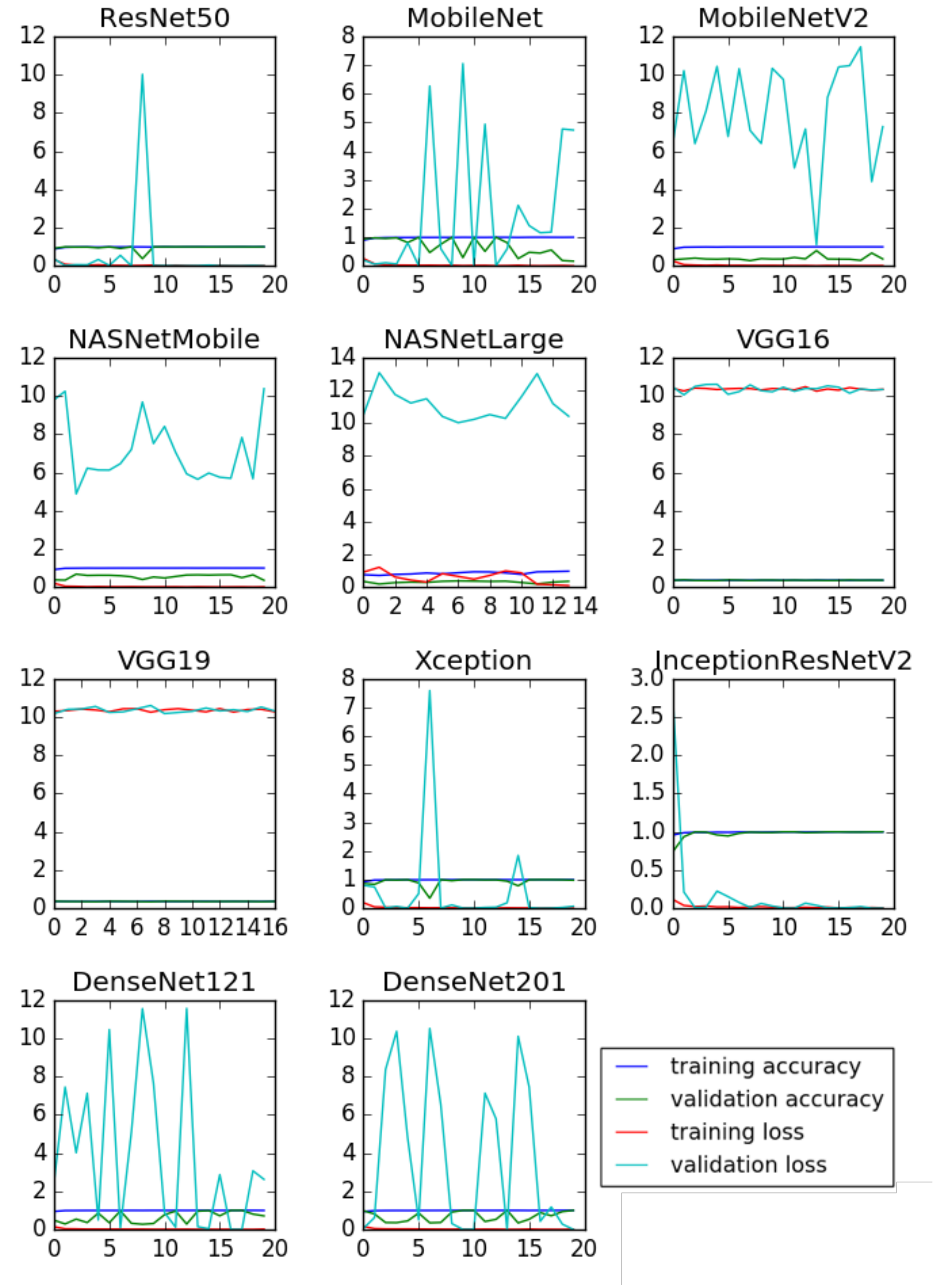


# Experimental setup

- Training Images: 16k
- Validation Images: 2k
- Testing Images: 4480

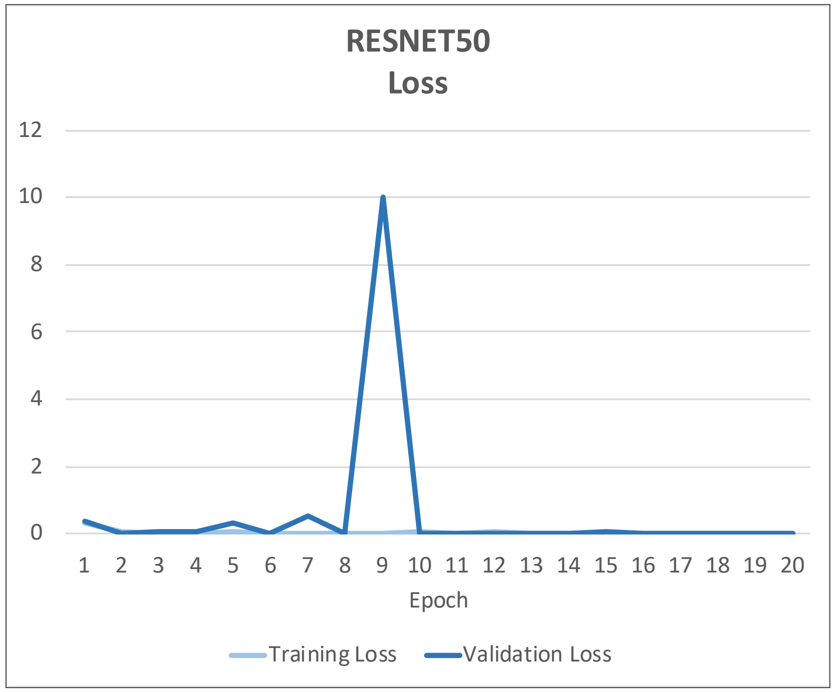
- Batch size: 16
- Epochs: 20

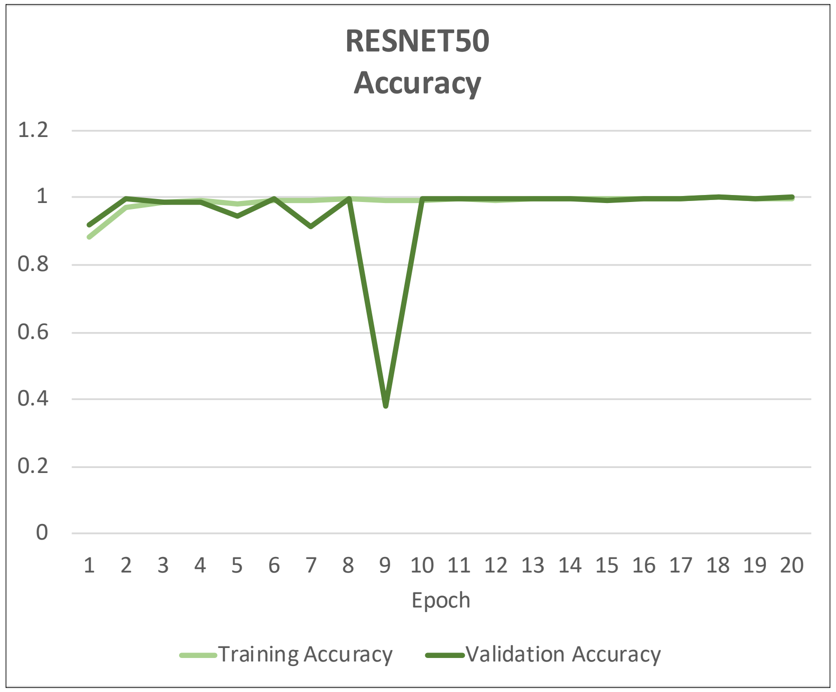


# Results

- Training Images: 16k
- Validation Images: 2k
- Testing Images: 4480

- Batch size: 16
- Epochs: 20



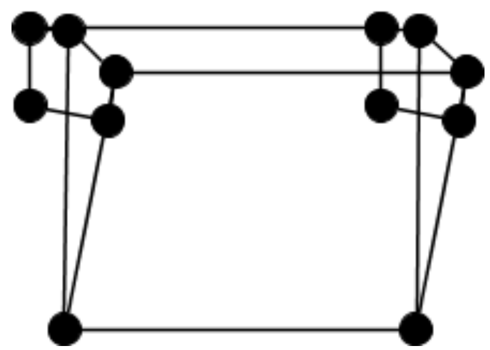




Best model:

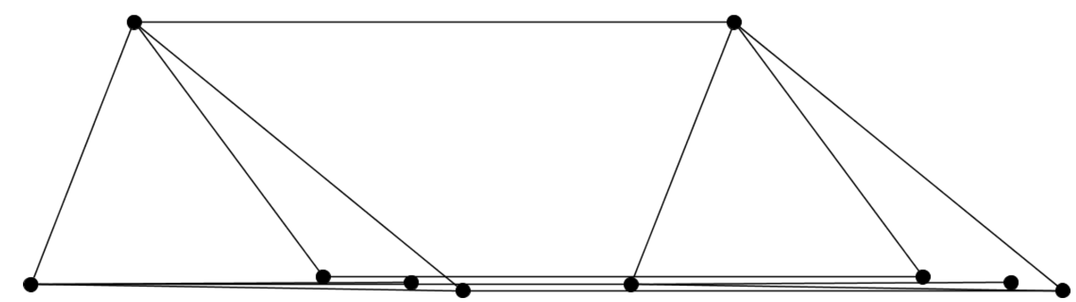
- **ResNet50**

**Accuracy: 99%**

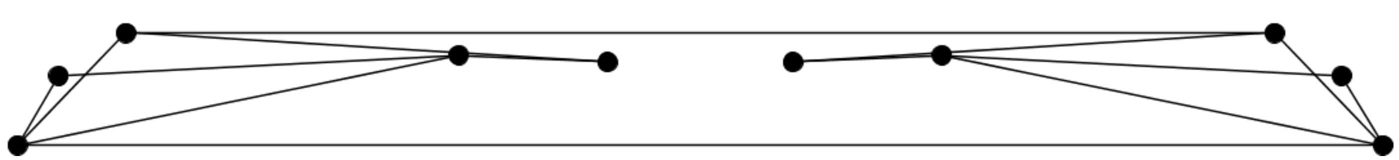
# Misclassified instance



~~Vertical~~ → Translational



~~Vertical~~ → Translational



~~Translation~~ → Vertical

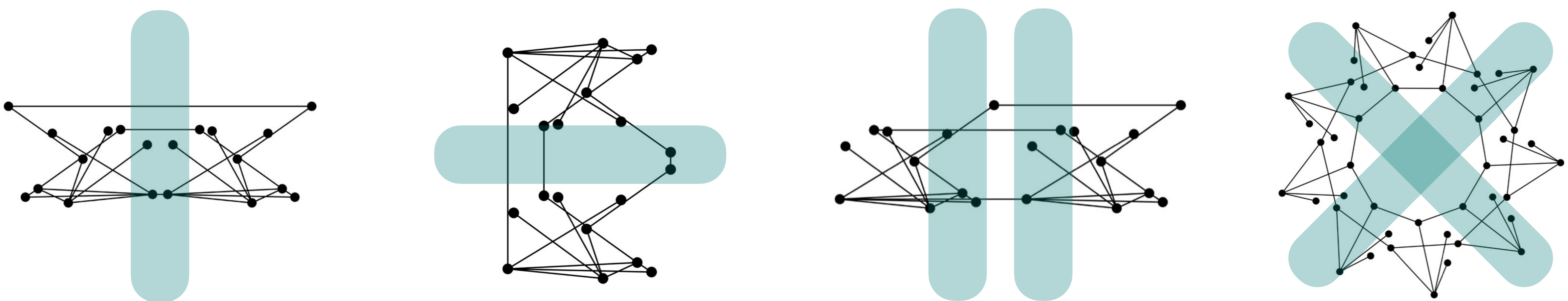
# Multi-class Classification:
# What type of symmetry does it show?

Reflectional

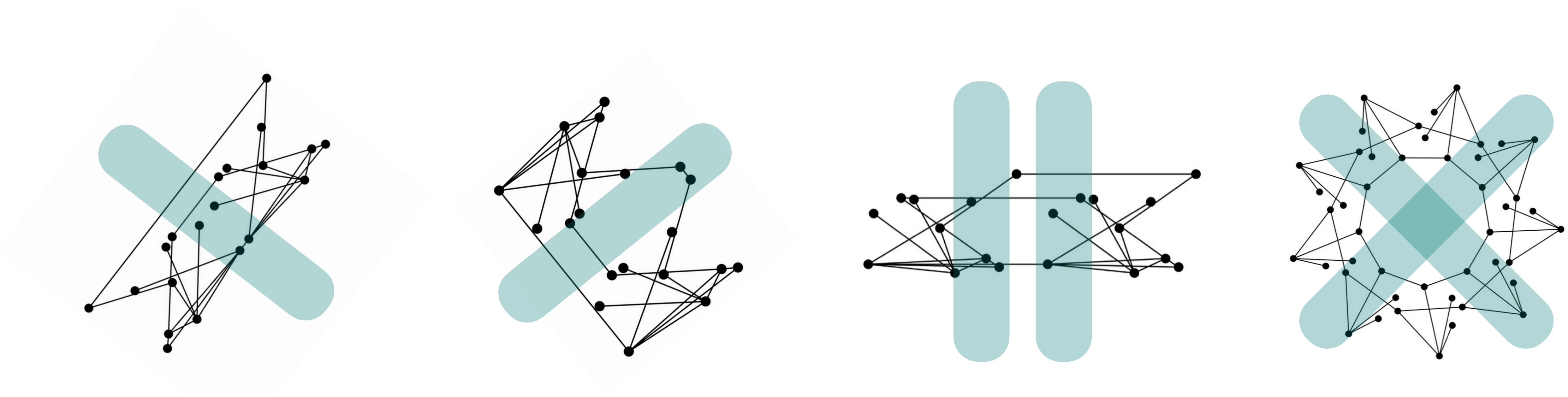
Translational

Rotational

# Types of layout

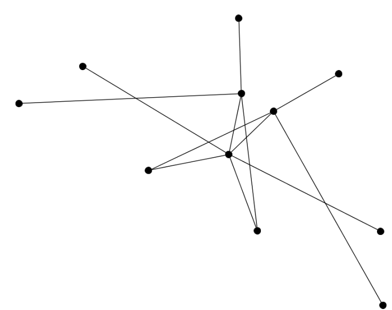
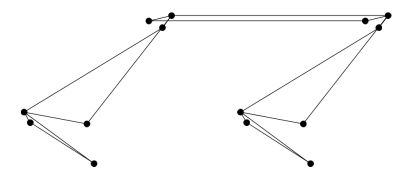
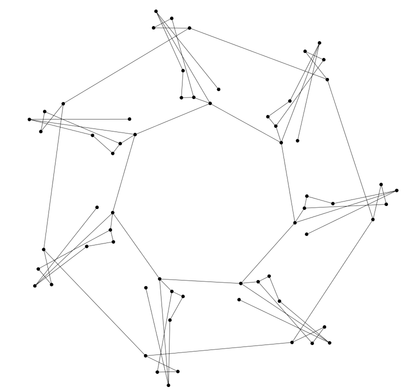


# Types of layout



Any axis of reflectional symmetry

# Results

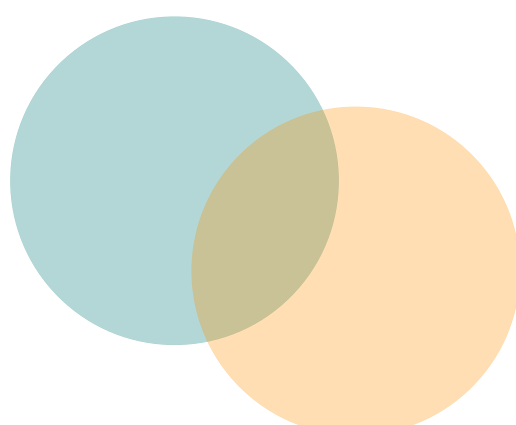




| | | Predicted | | |
|---|---|---|---|---|
| | | Reflectional | Translational | Rotational |
| Real | Reflectional | 872 | | |
| | Translational | 800 | | |
| | Rotational | | | 800 |

**Accuracy 69%**

# Conclusions

- We designed and implemented algorithms for generating symmetric and non-symmetric layouts
- We made available a dataset for training and testing machine-learning based approaches
- We used a CNN approach to detect and classify symmetries



Is symmetric?

- High accuracy for small graphs

What type of symmetry?

- High accuracy: H, V, T, R
- Lower accuracy: Ref, Tran, Rot

# Future Work

- Detect the angle of the main axis of symmetry in a graph drawing
- Evaluate the approach with respect to the size of the graph and the quality of image
- Improve the layout generation to encode additional characteristics
- Consider Graph Neural Networks that work with the structure of the graph
- Define a robust ML-based measure of symmetry in graph layouts
- Use such a measure to create more symmetric graph layouts

# Thank you!

18.09.2019 GD Morning Run 10k

**Felice De Luca**

felicedeluca@email.arizona.edu

CNN: https://github.com/enggiqbal/mlsymmetric

RAW layouts dataset: https://github.com/felicedeluca/mlsymdata

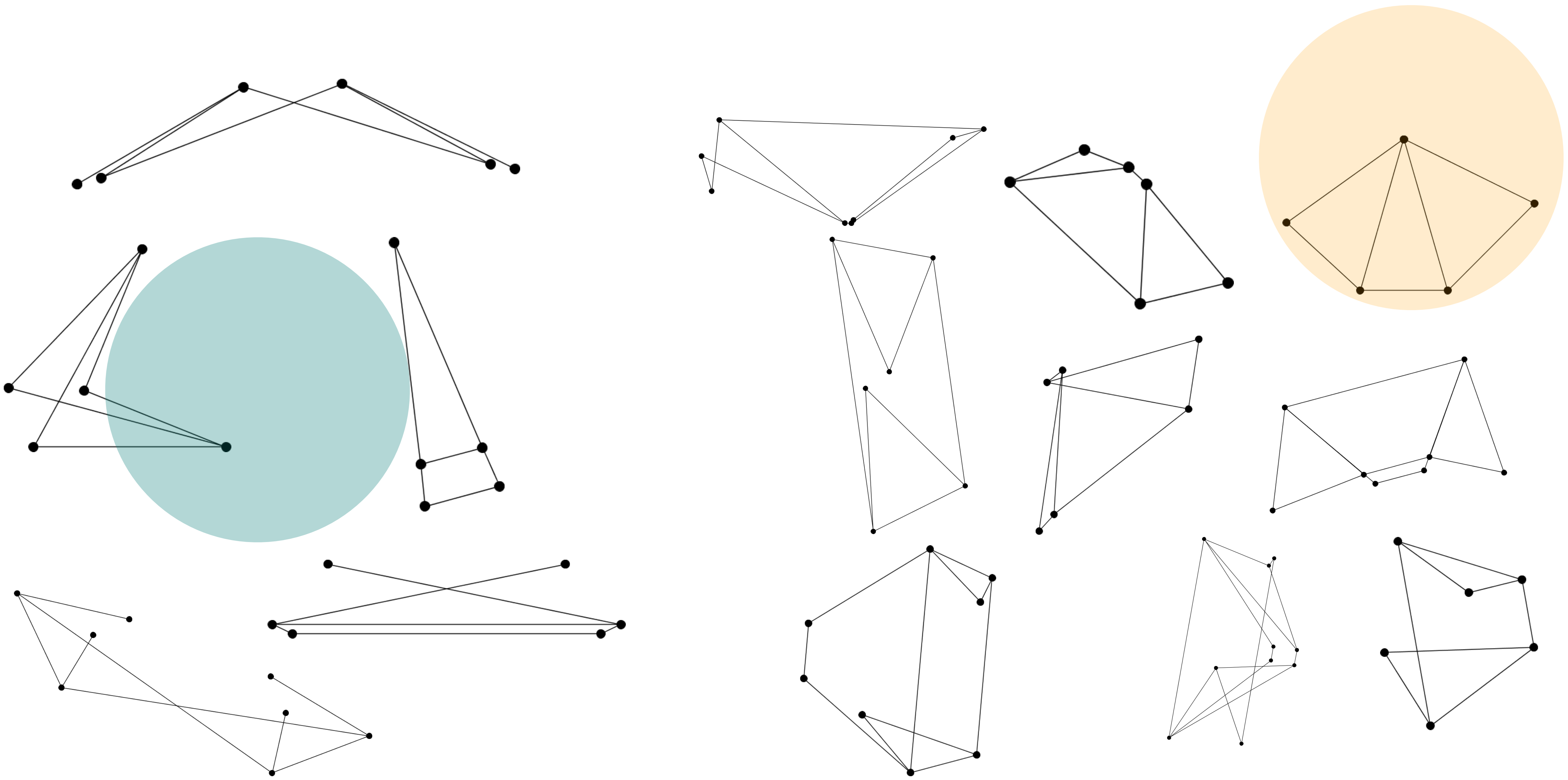
# Future Work

- Detect the angle of the main axis of symmetry in a graph drawing
- Evaluate the approach with respect to the size of the graph and the quality of image
- Improve the layout generation to encode additional characteristics
- Consider Graph Neural Networks that work with the structure of the graph
- Define a robust ML-based measure of symmetry in graph layouts
- Use such a measure to create more symmetric graph layouts

**Felice De Luca**
felicedeluca@email.arizona.edu

CNN: https://github.com/enggiqbal/mlsymmetric
RAW layouts dataset:
https://github.com/felicedeluca/mlsymdata

# Misclassified instance by K P CNN

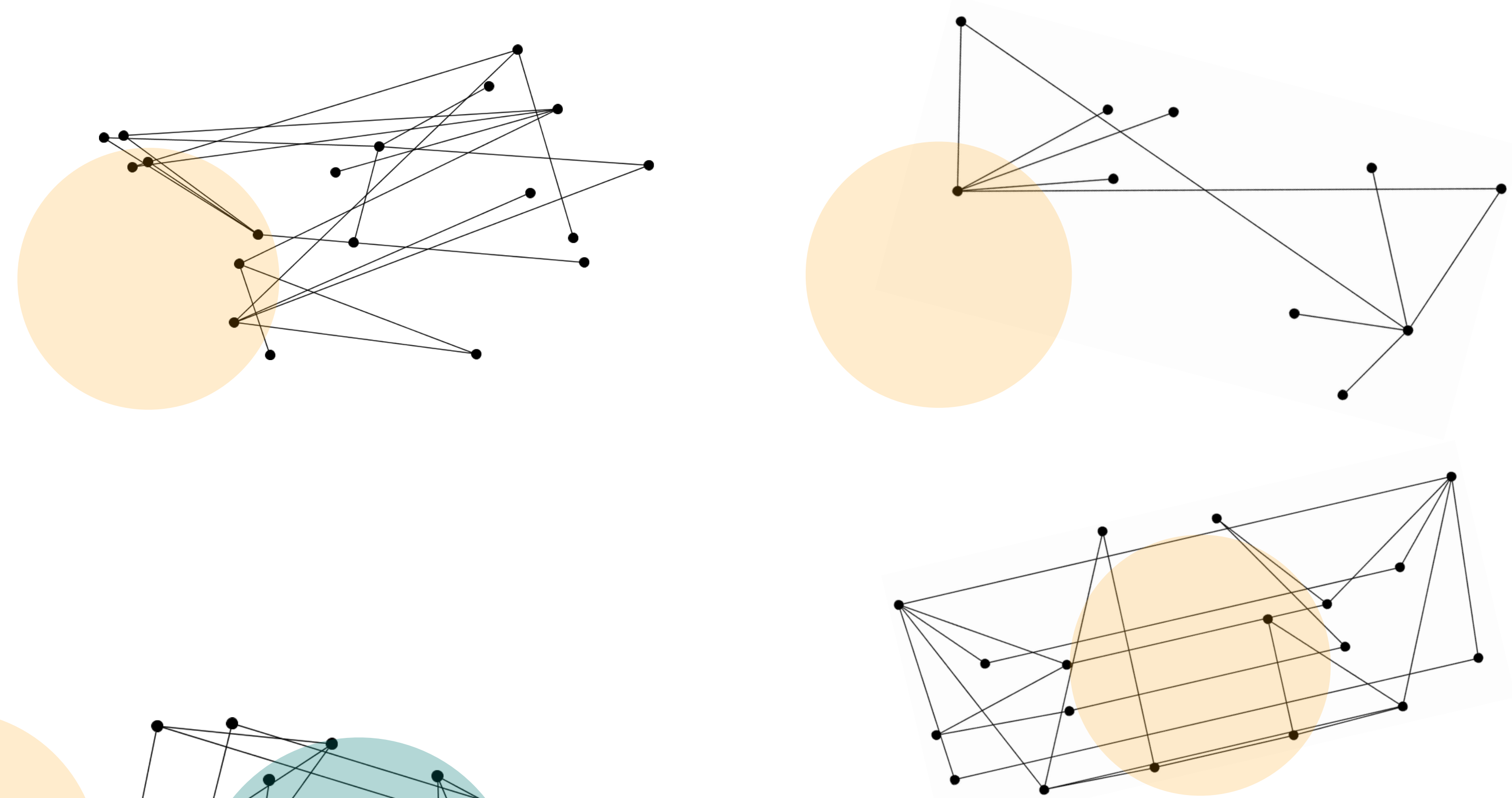


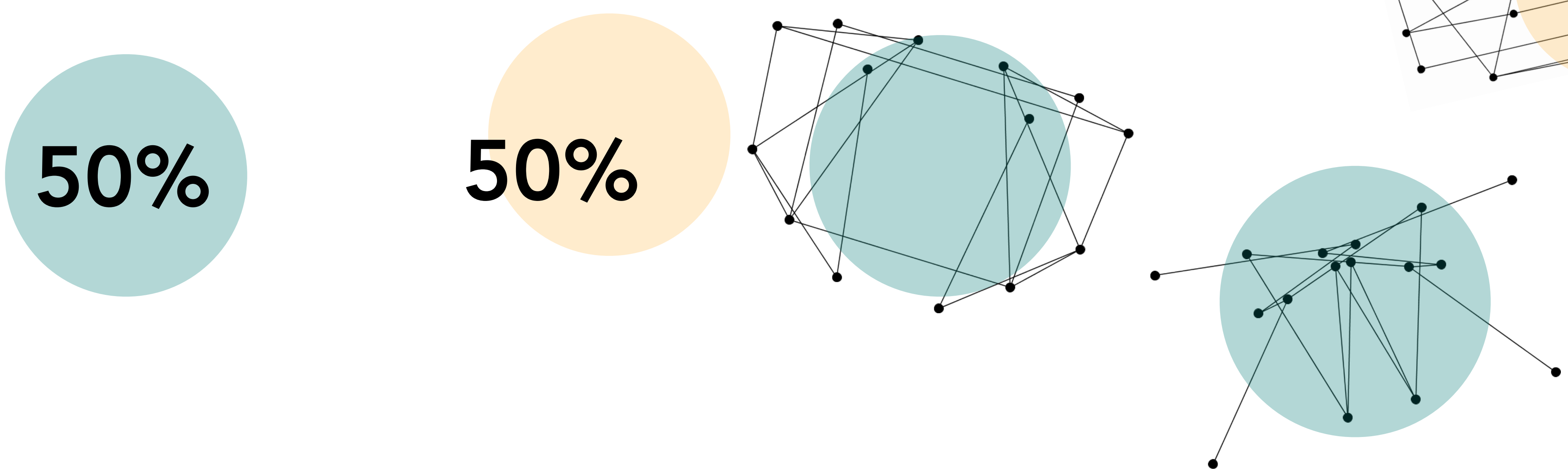
# Binary classification with bigger graphs

Vertices ∈ [10 , 20]

Edges ∈ [|V|, 1.2 * |V|]

Same setting



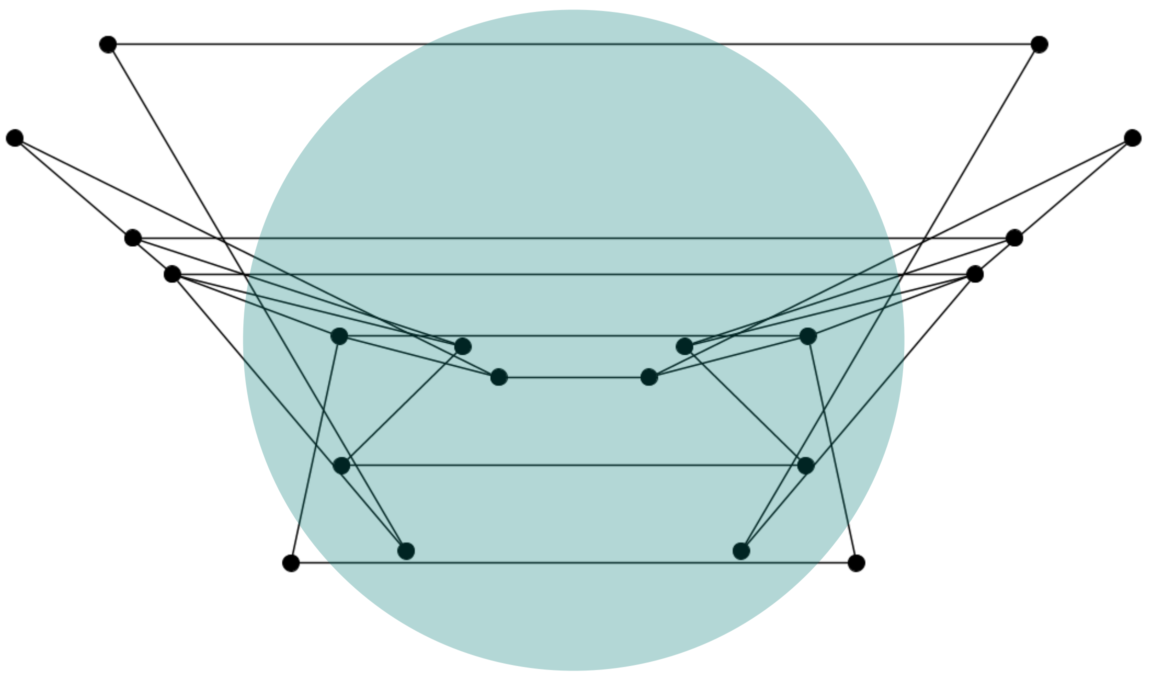



**Accuracy: 78%**

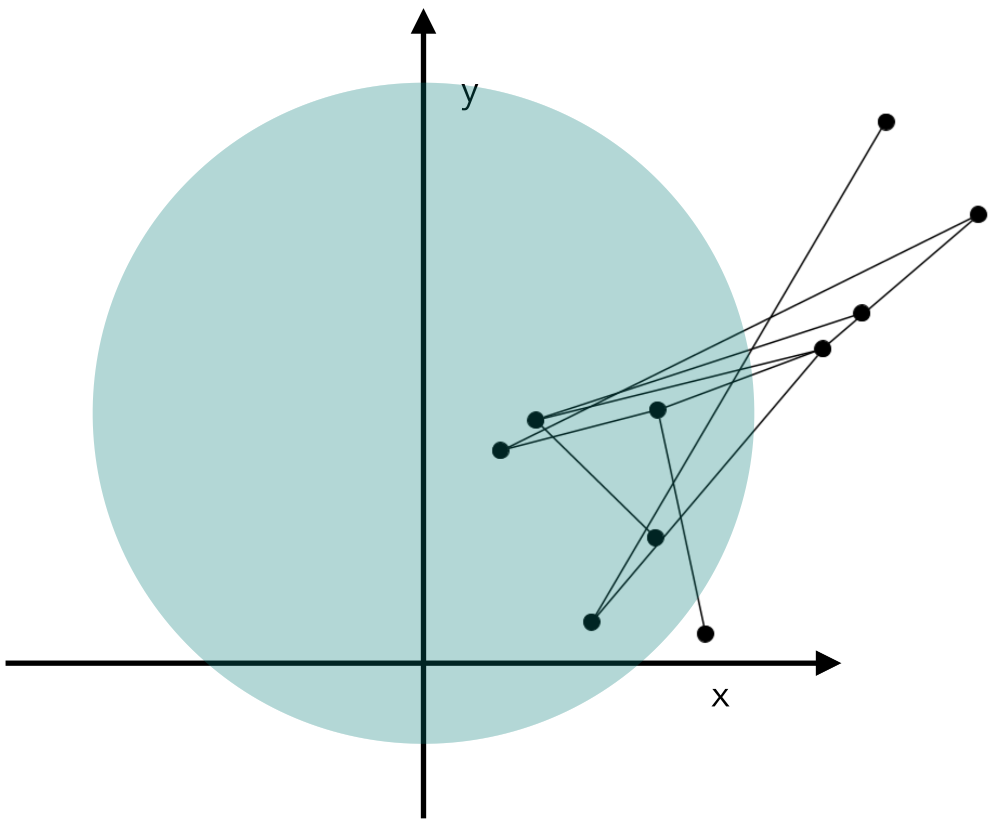
# Limitations

- Graphs are small
- Images are not too crowded
- Artificial graph layouts
- We only tested with 50-50 training and testing ratio
- No human baseline
- Lack of score of symmetricness for the layouts

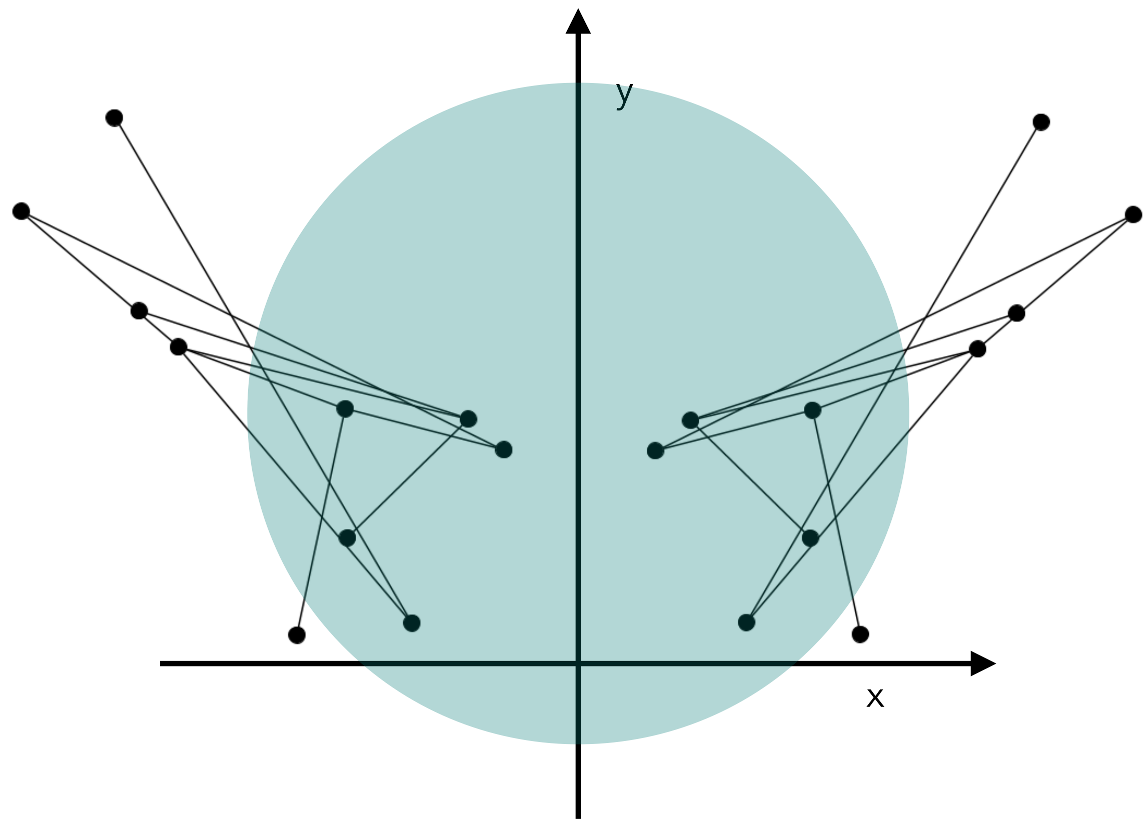
# Reflectional layout



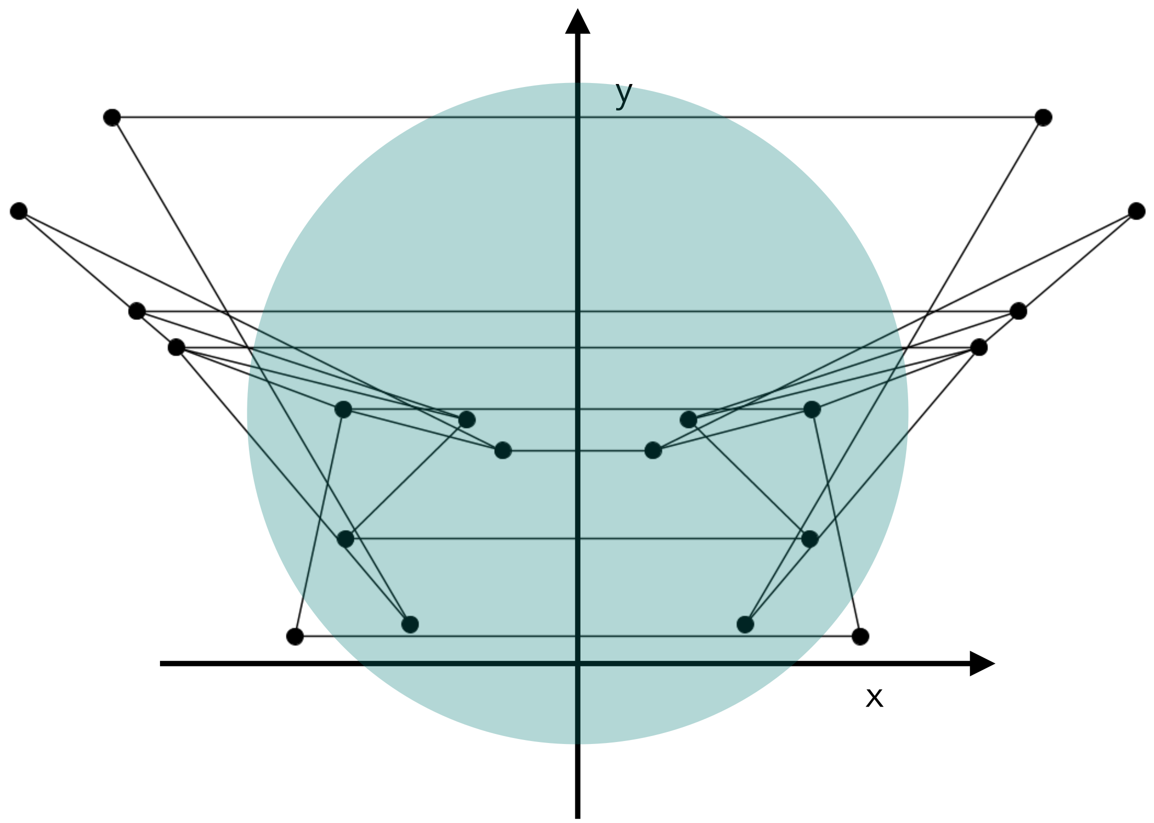
# Reflectional layout



# Reflectional layout



# Reflectional layout



# Symmetry detection in real world images

A competition to detect axis of symmetry took place in 2013 (Liu2013)

Loy and Eklundh (2006) won

Cicconet et al. (2016) detects also segment of symmetry



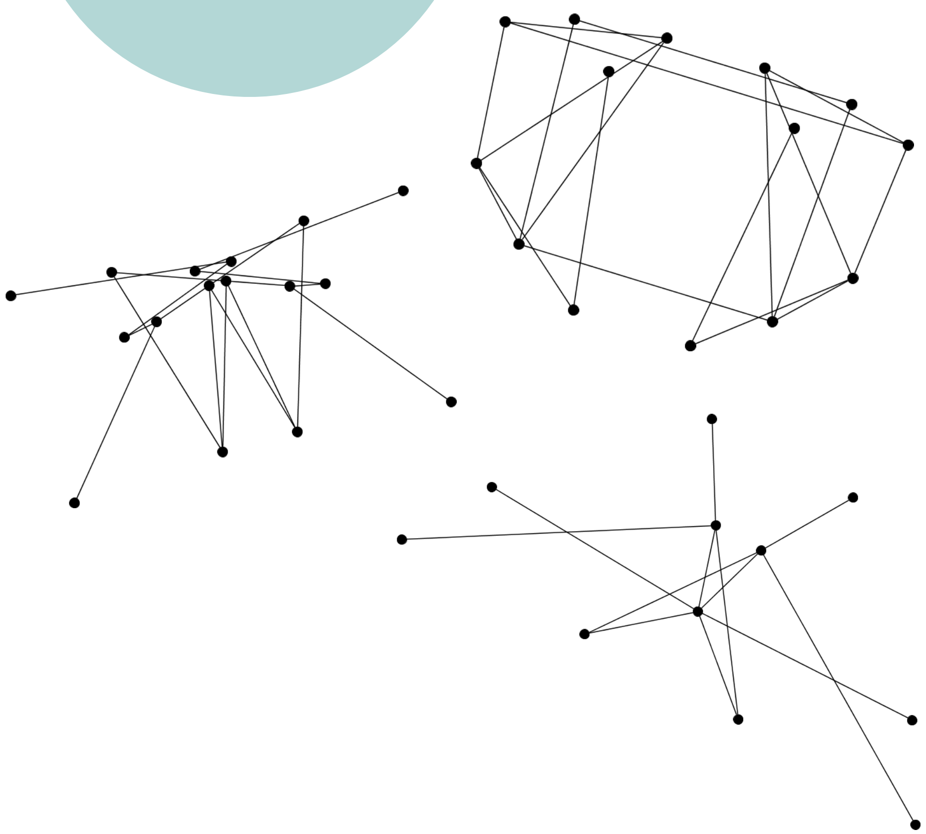


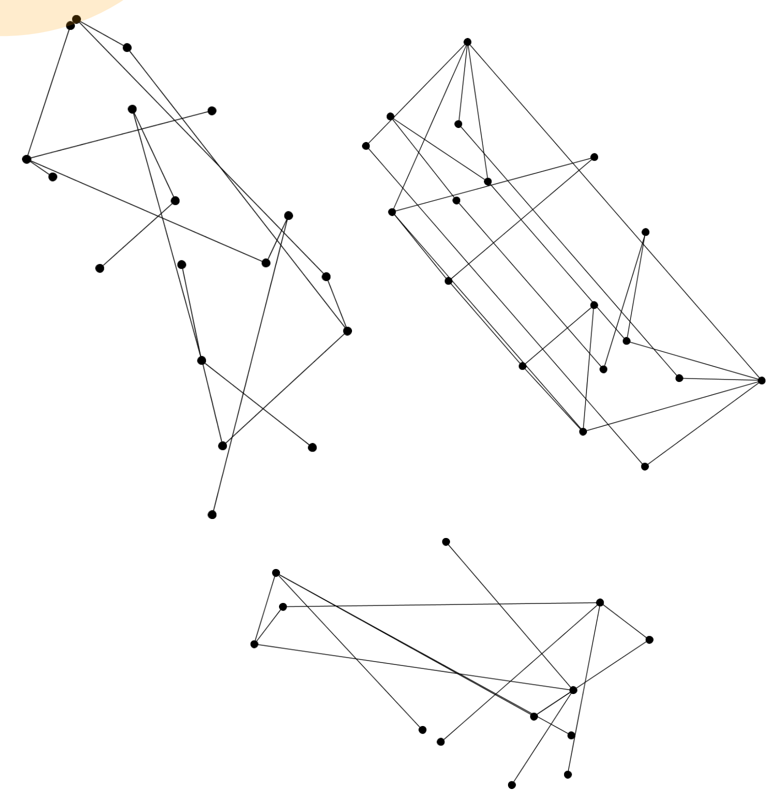

http://www.flickr.com/groups/symmetrycompetition

# Reflectional dataset

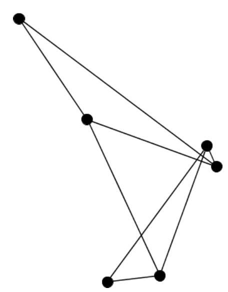
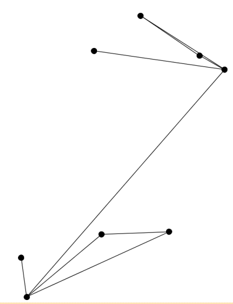
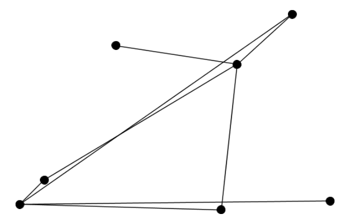
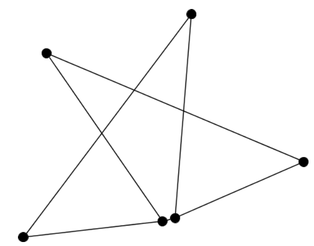
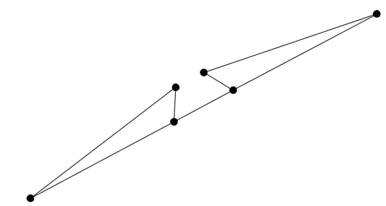
Symmetric

Non Symmetric



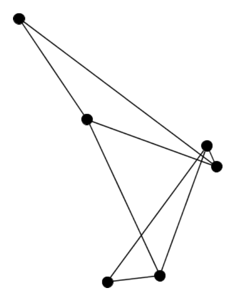
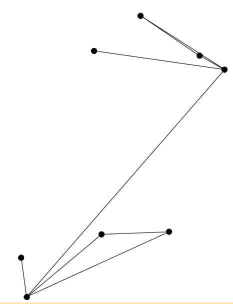
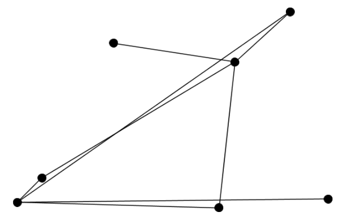
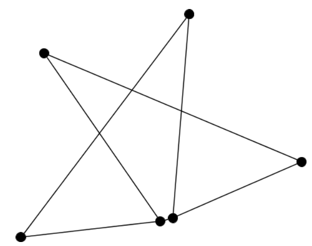
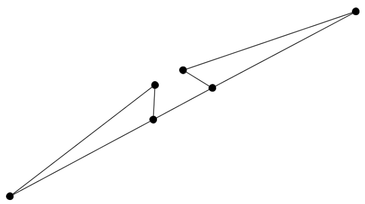


# Misclassified instances

| | PRandom crossings | PRandom parallel | Random | Symmetric crossings | Symmetric parallel |
|---|---|---|---|---|---|
| Our Approach | 26 | 43 | 32 | 36 | 14 |
| Klaupaukh | 71 | 96 | 48 | 131 | 9 |
| Purchase | 0 | 26 | 1 | 335 | 0 |

# Misclassified instances

| | PRandom crossings | PRandom parallel | Random | Symmetric crossings | Symmetric parallel |
|---|---|---|---|---|---|
| Our Approach | 26 | 43 | 32 | 36 | 14 |
| Klaupaukh | 71 | 96 | 48 | 131 | 9 |
| Purchase | 0 | 26 | 1 | **335** | 0 |

# Trained CNN models

Keras

**models we used**

| | | | |
|---|---|---|---|
| ResNet50 | MobileNet | MobileNetV2 | NASNetMobile |
| NASNetLarge | VGG16 | VGG19 | Xception |
| InceptionResNetV2 | DenseNet121 | DenseNet201 | |