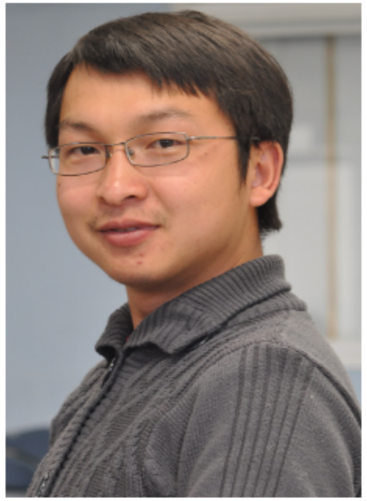
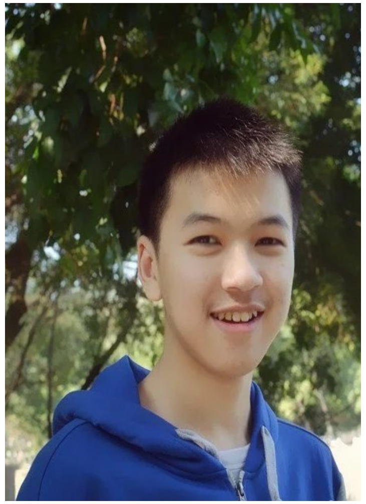
DeepDrawing: A Deep Learning Approach to Graph Drawing



Yong Wang[1]. Zhihua Jin[1,4] Qianwen Wang[1] Weiwei Cui[2]. Tengfei Ma[3]. Huamin Qu[1]

http://yong-wang.org/proj/deepDrawing.html

1 香港科技大學 THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

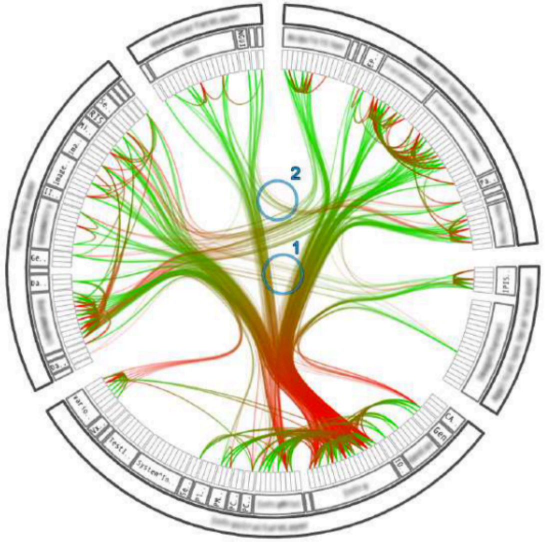
2 Microsoft Research 微软亚洲研究院

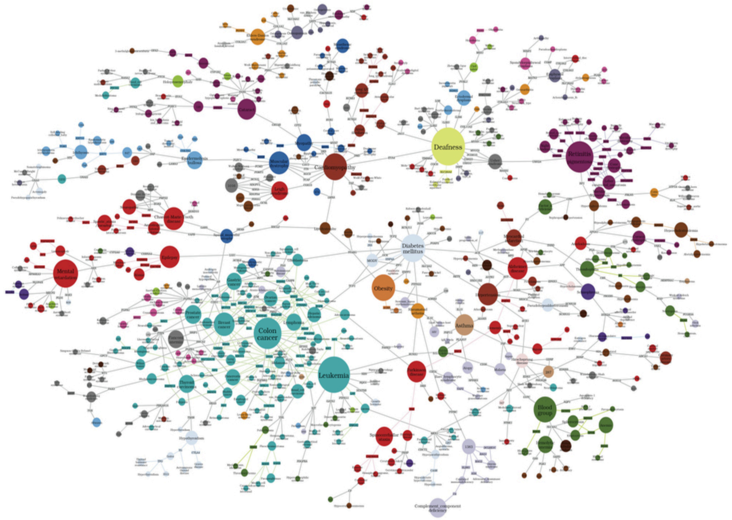
3 IBM

4 浙江大学 ZHEJIANG UNIVERSITY 1897

Motivation

- Graph drawing has been extensively studied to facilitate the exploration, analysis and presentation of networks!



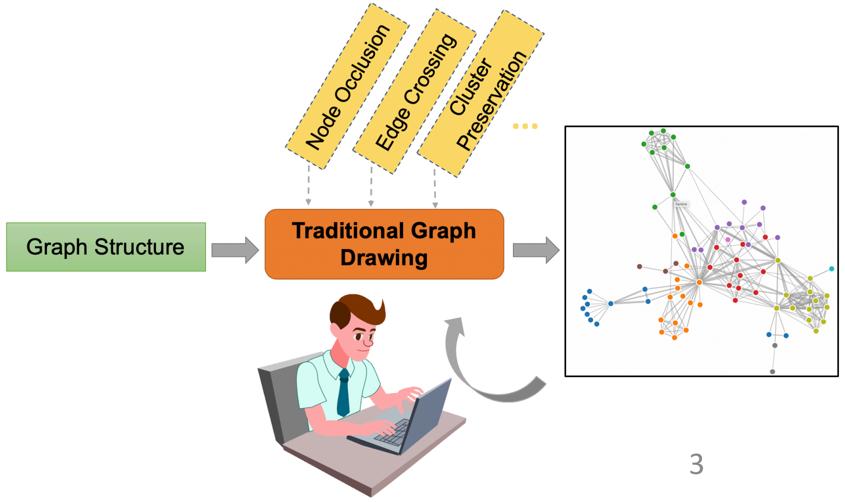


Motivation

- Graph drawing has been extensively studied to facilitate the exploration, analysis and presentation of networks!

- However, users often need to find a desirable graph layout through trial-and-error:
  - Tune different algorithm-specific parameters

  - Compare different drawing results



Motivation

- Graph drawing has been extensively studied to facilitate the exploration, analysis and presentation of networks!

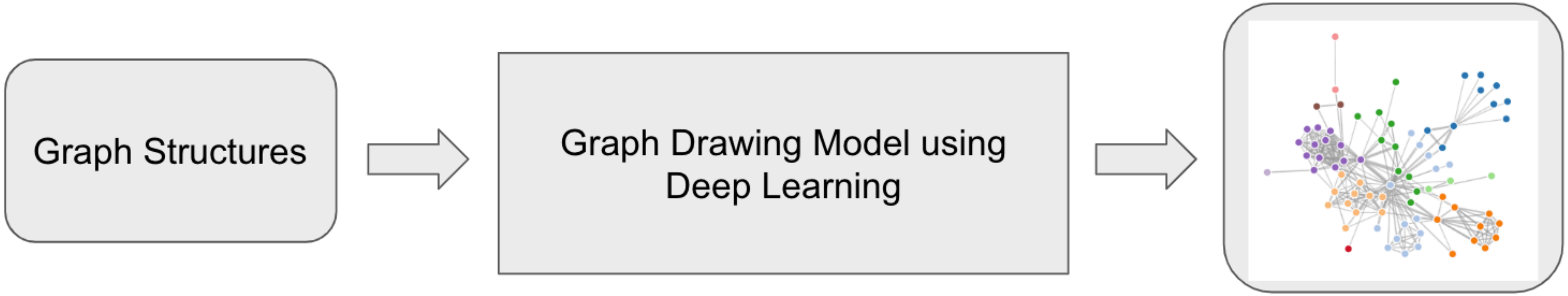
- However, users often need to find a desirable graph layout through trial-and-error:

It is time-consuming and not user-friendly, especially for non-expert users!

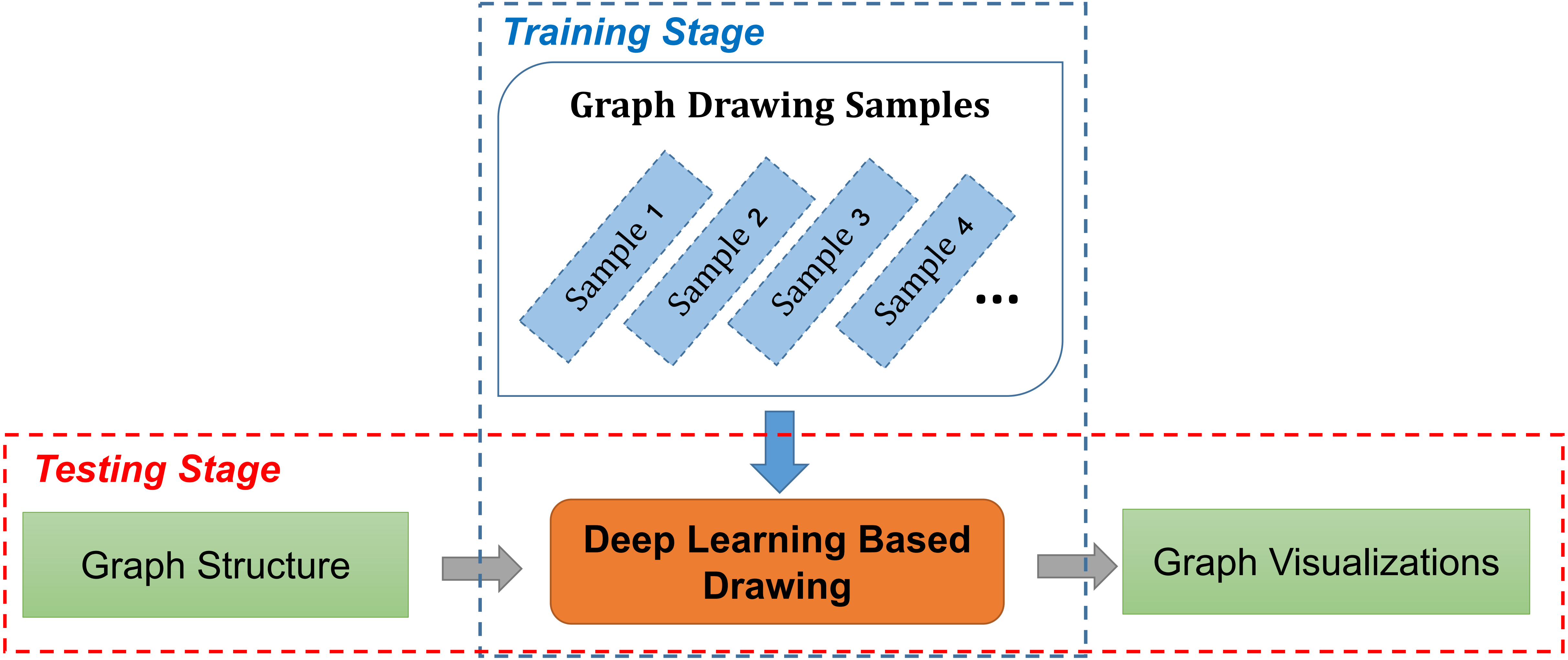
Research Question

- Deep learning techniques have shown a powerful capability of modelling the training data and further making predictions in many applications

- Can we model graph drawing as a learning and prediction problem and further generate drawings for input graphs directly?



Overall Idea



Challenges

- **Model Architecture**
- **Loss Function Design**
- **Training Datasets**

Challenges

- **Model Architecture**

  - Existing deep learning techniques are mainly applied to the Euclidean data (e.g., images, videos and texts), instead of graphs

  - Recent research on Graph Neural Network mainly targets at node classification and link prediction on **a single graph**, which is much different from graph drawing

Challenges

- **Model Architecture**

- **Loss Function Design**

  - How to evaluate whether a drawing for an input graph is ”correct” or not?

Challenges

- **Model Architecture**
- **Loss Function Design**
- **Training Datasets**
  - There are no publicly-available high-quality datasets for graph drawing

DeepDrawing

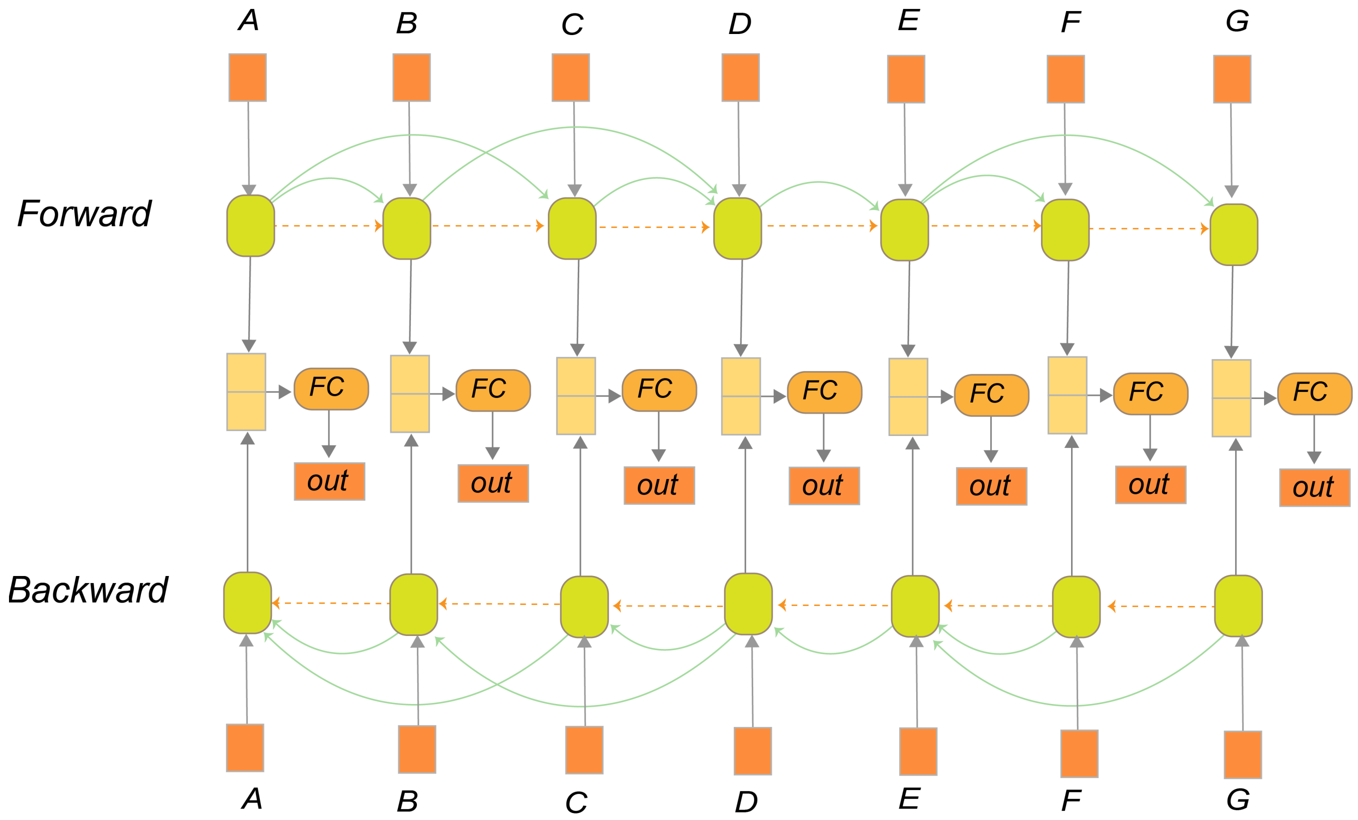
- Model Architecture
- Model Input
- Loss Function Design
- Dataset Generation

DeepDrawing – Model Architecture

- Major Considerations
  - The majority of graph neural networks mainly focus on the learning and prediction tasks for a single graph
  - However, a recent study[1] has shown that RNNs are capable of modelling the structure information of multiple graphs

[1] J. You, R. Ying, X. Ren, W. L. Hamilton, and J. Leskovec. Graphrnn: a deep generative model for graphs. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

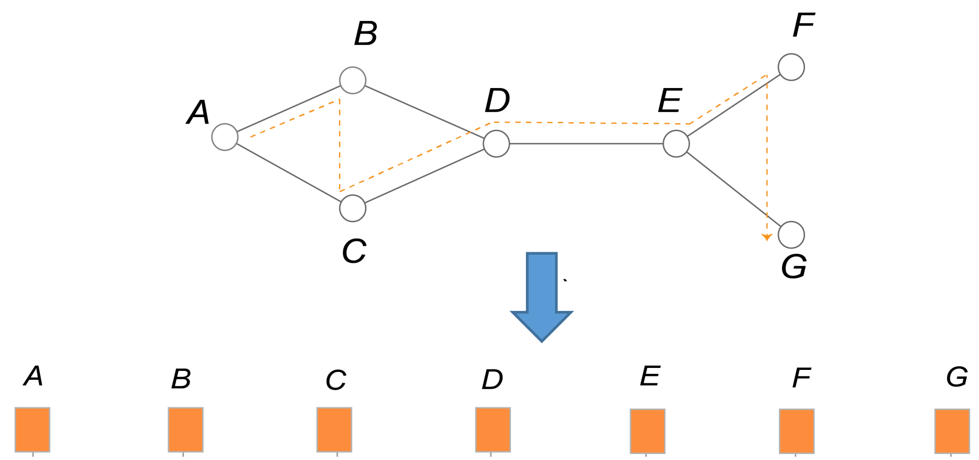
DeepDrawing – Model Architecture



We propose a bi-directional graph-LSTM based model for graph drawing.

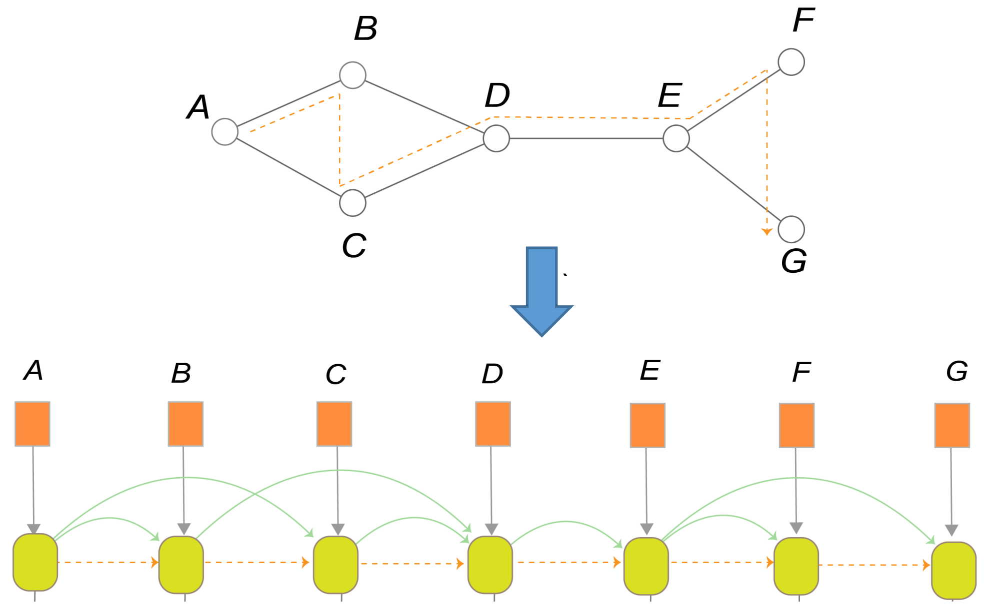
DeepDrawing – Model Architecture

- Architecture Details:
  - BFS-ordering of graph nodes



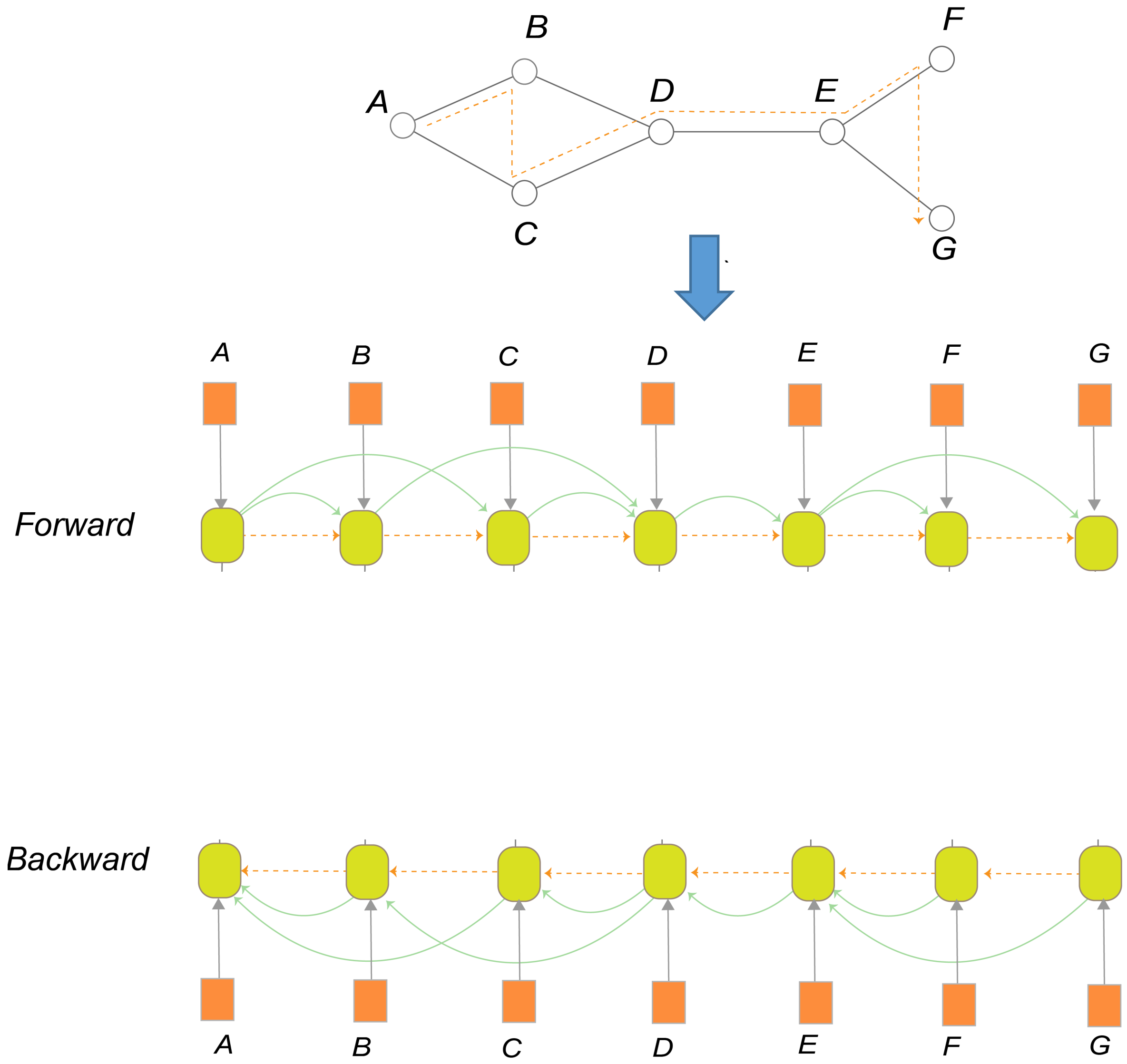
DeepDrawing – Model Architecture

- Architecture Details:
  - BFS-ordering of graph nodes
  - Fake edges (dotted yellow arrow) and real edges (green arrow)



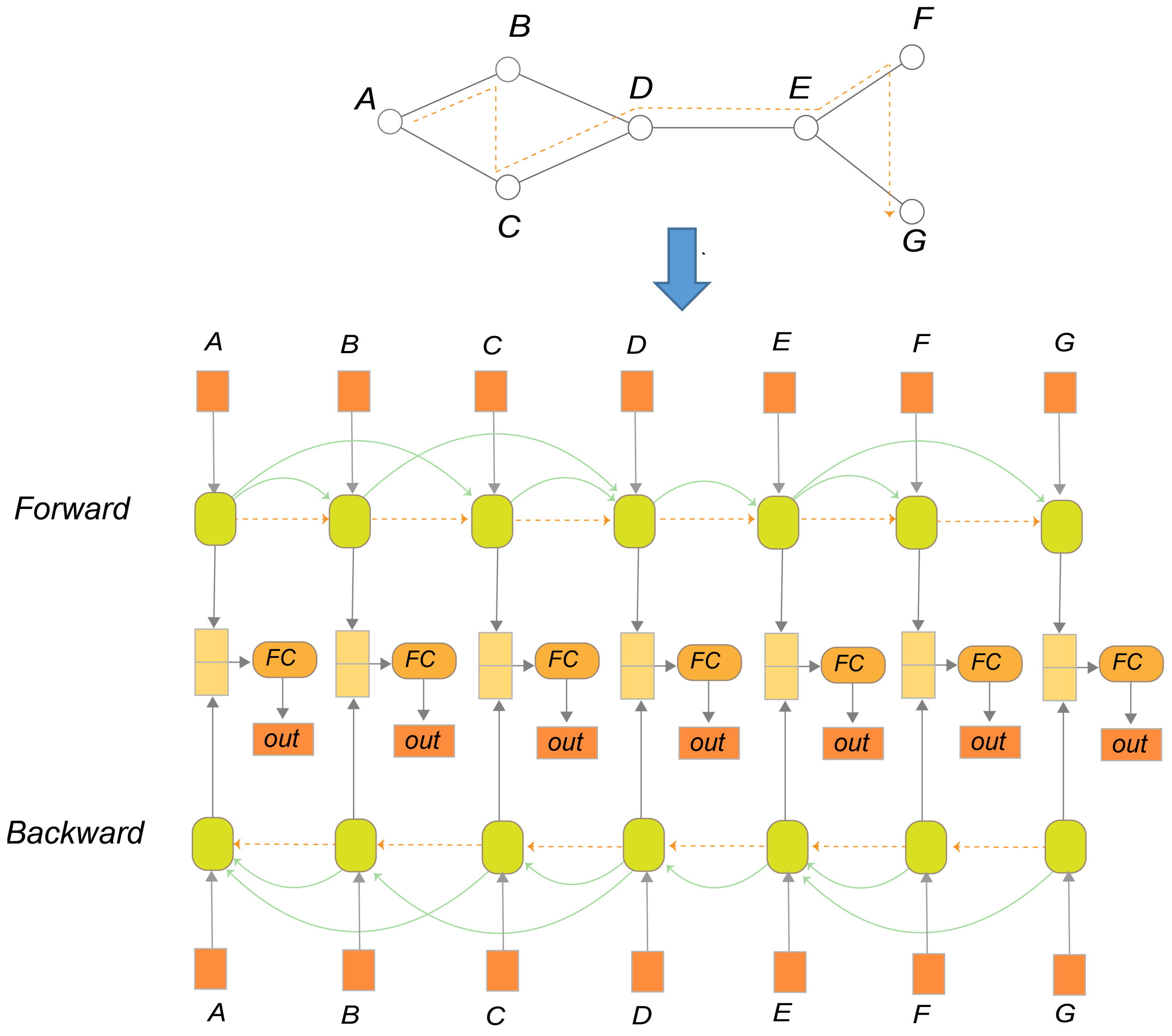
DeepDrawing – Model Architecture

- Architecture Details:
  - BFS-ordering of graph nodes
  - Fake edges and real edges
  - Bi-directional



DeepDrawing – Model Architecture

- Architecture Details:
  - BFS-ordering of graph nodes
  - Fake edges and real edges
  - Bi-directional



DeepDrawing – Model Input

- Node Feature Vector
  - Natural choice: node embedding

They mainly target at single graphs and are not able to be generalized to multiple graphs[2]!

  - **A fixed-length adjacency vector** encoding the connection information between the current node and its prior nodes.

[2] M. Heimann and D. Koutra. On generalizing neural node embedding methods to multi-network problems. In *KDD MLG Workshop*, 2017.

DeepDrawing – Model Input

- Node Ordering
  - Random ordering

    The possible orderings for an input graph can be very large!

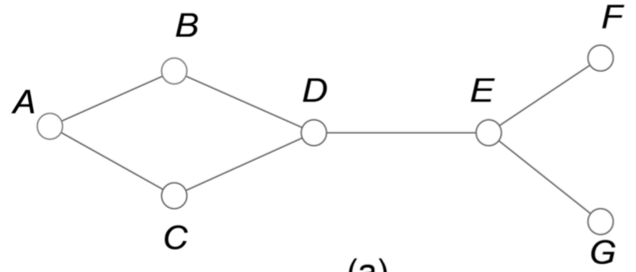
  - **BFS ordering**

    - Avoid exhaustively going through all possible node permutations

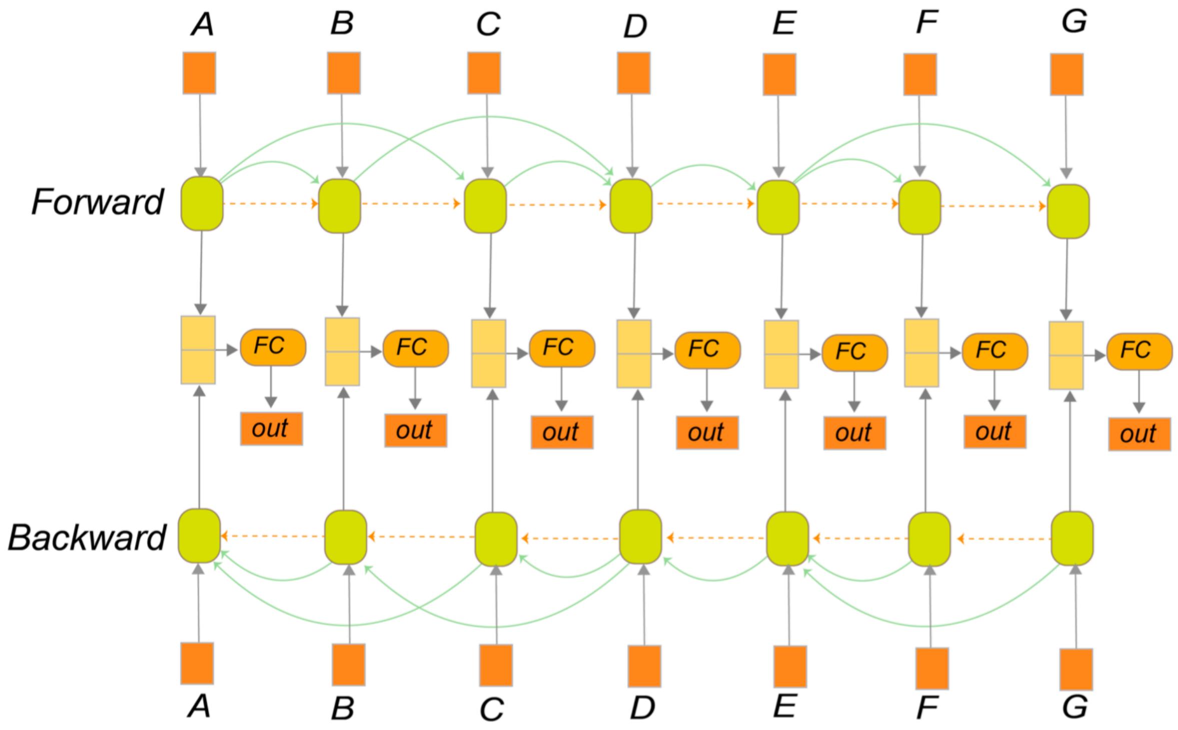
    - There is **an upper bound** for the possible connection between the current node and its prior furthest nodes along the BFS sequence[1]!

[1] J. You, R. Ying, X. Ren, W. L. Hamilton, and J. Leskovec. Graphrnn: a deep generative model for graphs. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

DeepDrawing – Model Input



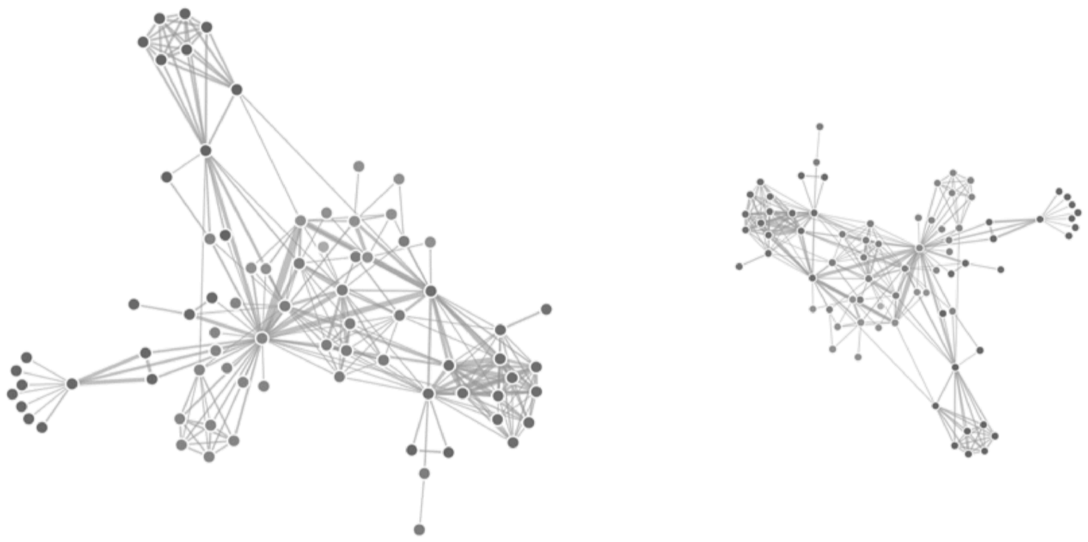
(a)



(b)

DeepDrawing – Loss Function Design

- Design Considerations
  - Make the predicted drawings as <u>**similar**</u> as possible to the drawings of ground-truth
  - The function should be invariant to translation, rotation and scaling



(a) (b)

DeepDrawing – Loss Function Design

- Procrustes Statistic

$$R^2 = 1 - \frac{(tr(C^T \bar{C}\bar{C}^T C)^{1/2})^2}{tr(C^T C)tr(\bar{C}^T \bar{C})}$$

  - It is transformation-invariant
  - It is between 0 and 1
  - Zero means the drawings are exactly the same; while one means they are totally different

DeepDrawing – Dataset Generation

- We generate:
  - Graph data: grid graphs, star graphs, clustered general graphs

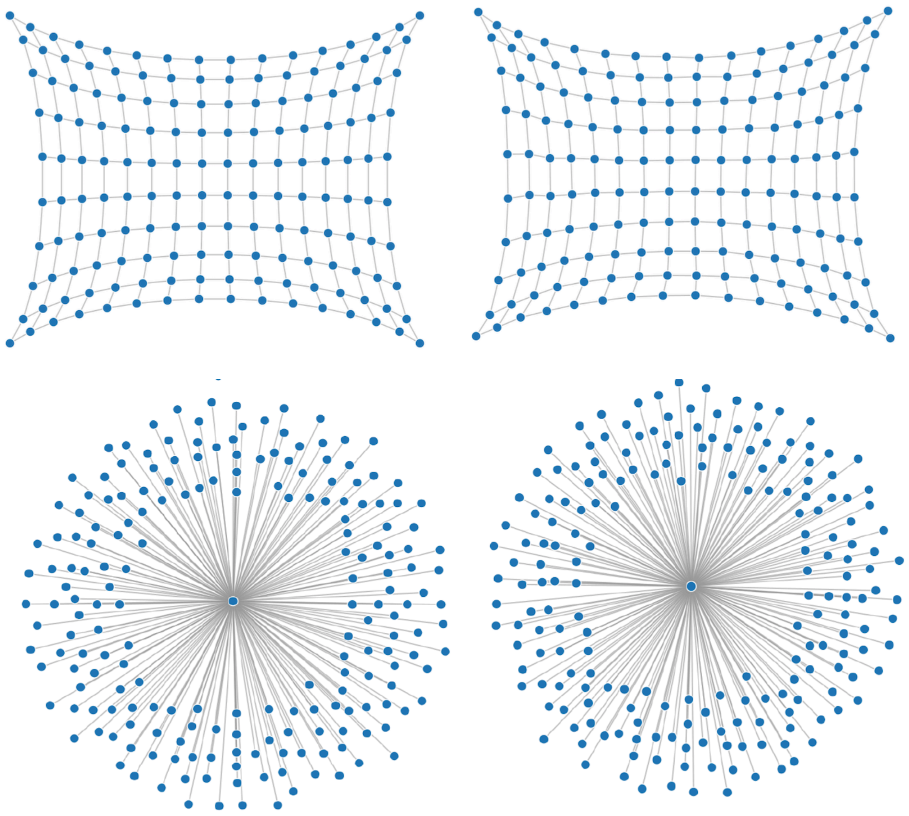
  - Graph drawing data: grid layout, star layout, ForceAtlas2, PivotMDS
    - We manually tune the parameters of the drawing algorithms

Evaluations

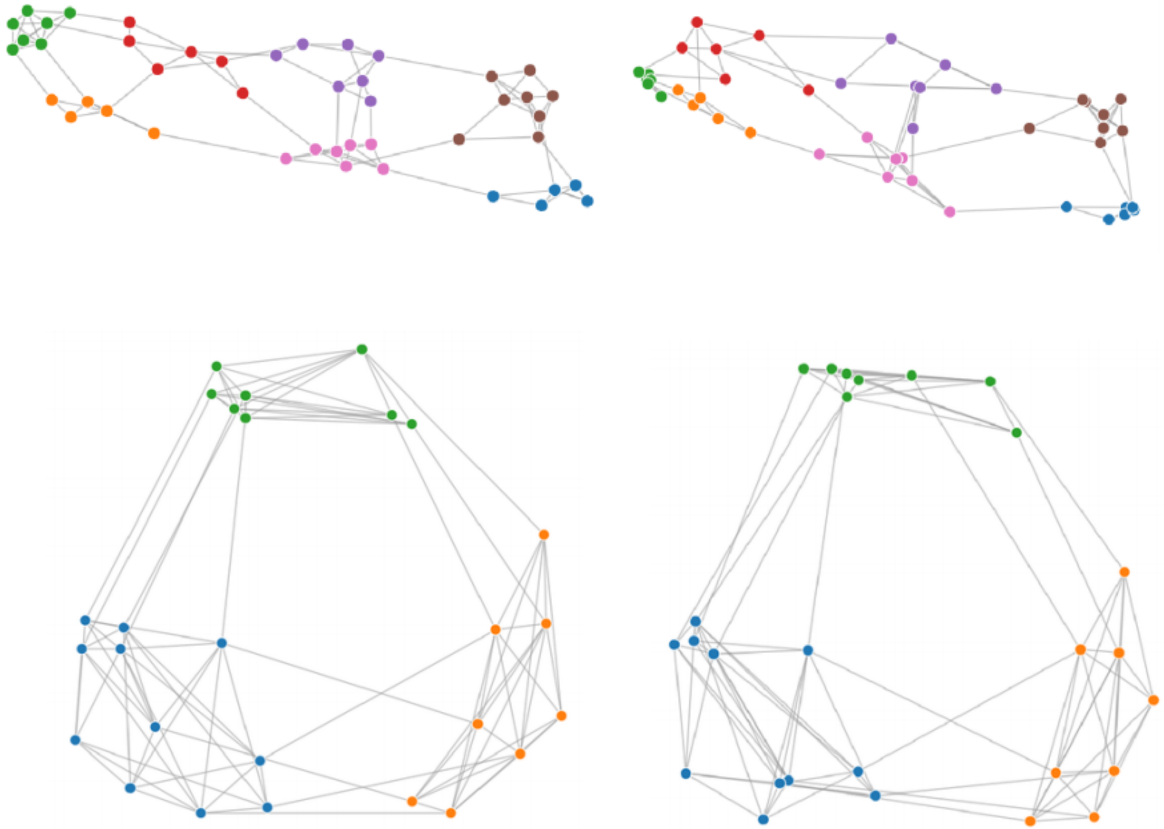
- We extensively evaluated the proposed approach:
  - Qualitative and quantitative evaluations
  - Comparison with the graph truth drawings and those by the baseline model (a 4-layer Bi-LSTM model)

Evaluations – Qualitative Evaluation

Ground-Truth Our Approach

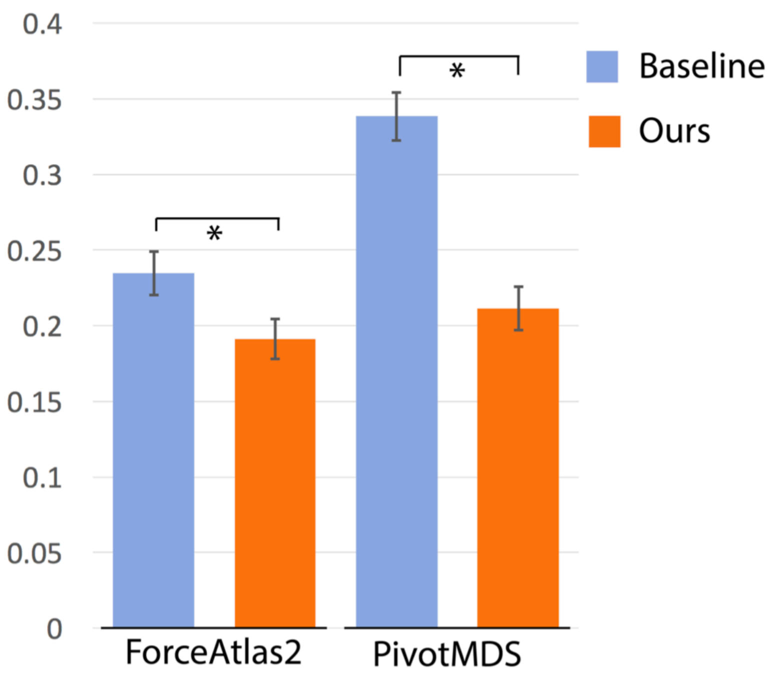


Ground-Truth Our Approach



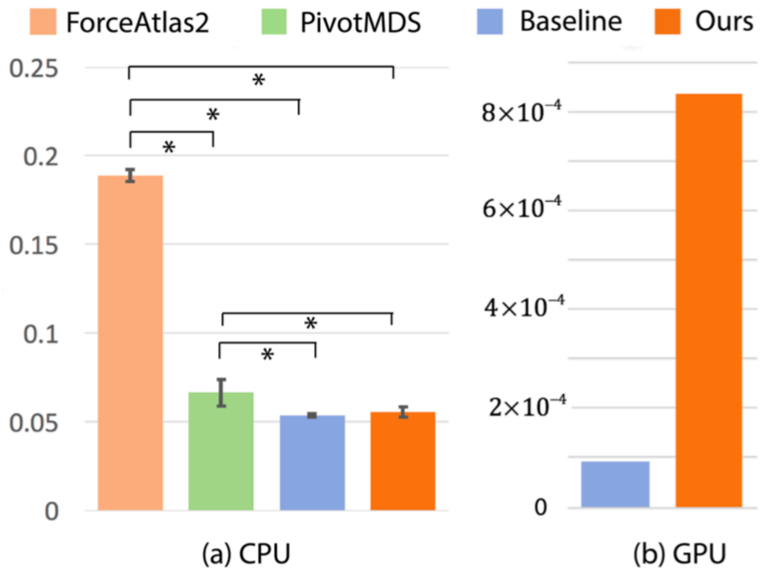
Evaluations – Quantitative Evaluation

- Procrustes Statistic-based similarity:
  Our approach is significantly better than the baseline model



Evaluations – Quantitative Evaluation

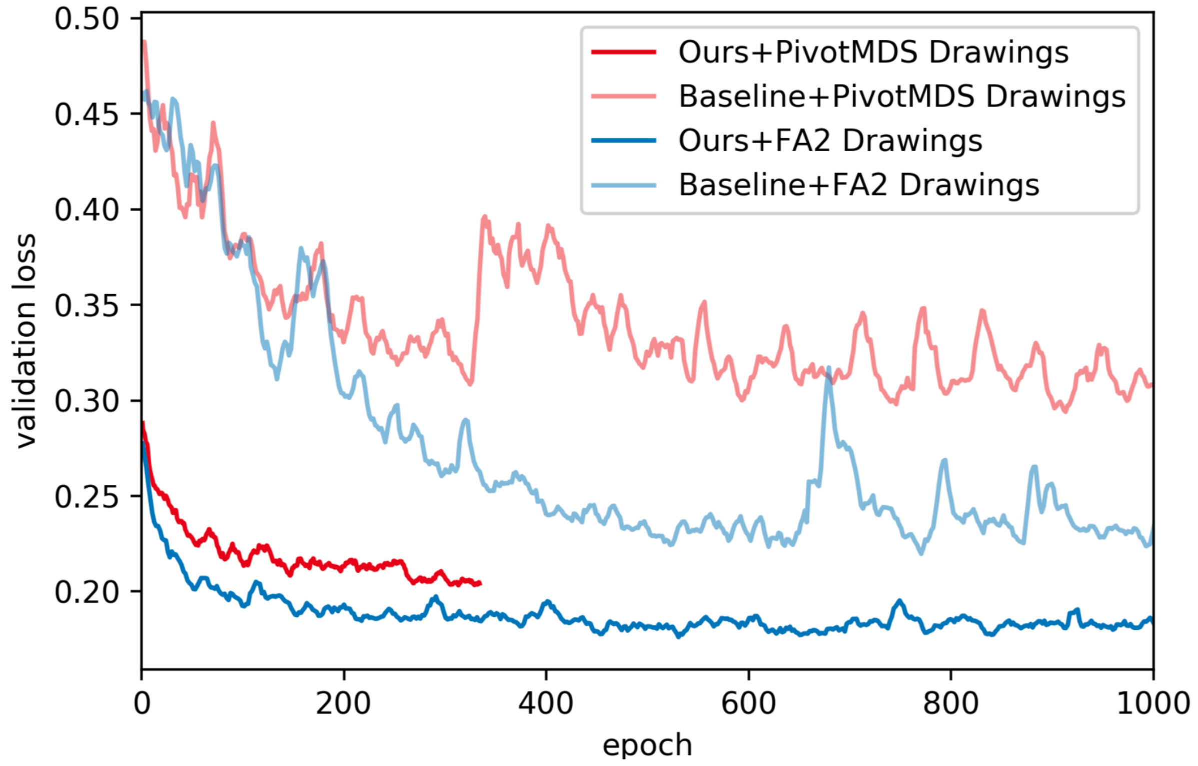
- Running Speed
  - CPU: Both our approach and the baseline model is faster than the traditional graph drawing methods
  - GPU: Our approach is slower than the baseline model on GPU, though it has 80% less parameters



Evaluations – Quantitative Evaluation

- Training Convergence Comparison

  Our approach can converge faster than the 4 layer Bi-LSTM in terms of #Epochs.



Limitations

- Lack Interpretability
- Our current evaluations mainly focus on small graphs with 20 to 50 nodes
- The performance of DeepDrawing has a dependence on the input node ordering and the structure similarity with the training graphs

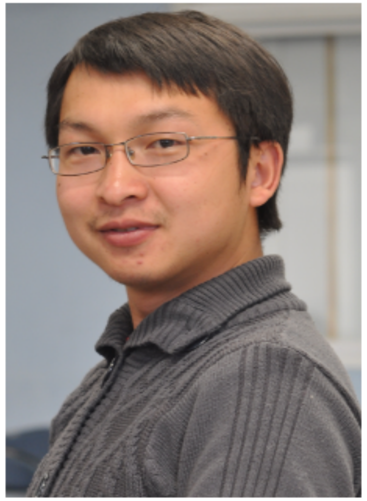
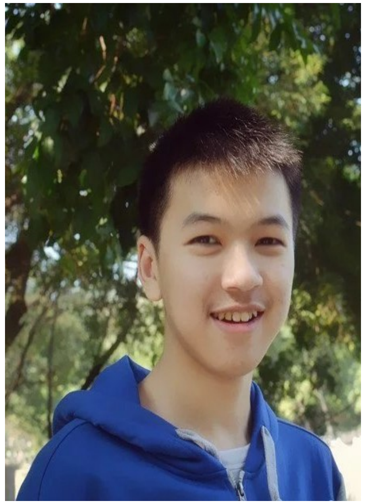
Take Home Message

- We propose a graph-LSTM based approach to graph drawing and investigate its effectiveness on small graphs

- It is worth further exploration in terms of good interpretability and better prediction performance on large graphs

- More details: code, video and slides are(or will be) accessible at:
  http://yong-wang.org/proj/deepDrawing.html

DeepDrawing: A Deep Learning Approach to Graph Drawing



Yong Wang[1.] Zhihua Jin[1,4.] Qianwen Wang[1] Weiwei Cui[2.] Tengfei Ma[3.] Huamin Qu[1]

http://yong-wang.org/proj/deepDrawing.html

1 香港科技大學 THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

2 Microsoft Research 微软亚洲研究院

3 IBM

4 浙江大学 ZHEJIANG UNIVERSITY 1897