Computational Intelligence and
Machine Learning Group

Deep Graph Networks

DAVIDE BACCIU (DAVIDE.BACCIU@DI.UNIPI.IT)

DIPARTIMENTO DI INFORMATICA - UNIVERSITA' DI PISA

IEEE TASK FORCE ON LEARNING FOR STRUCTURED DATA

www.learning4graphs.org

Lectures Outline

- Part I - Fundamentals
  - An introduction to learning with graphs
  - Contractive and contextual graph processing
  - Quick literature survey
- Part II - Generative approaches and research directions
  - Learning with generative models and learning to generate graphs
  - Advanced topics, research directions and reproducibility
  - Applications

Computational Intelligence and Machine Learning Group

ACDL 2021

Deep Graph Networks - Fundamentals

DAVIDE BACCIU (DAVIDE.BACCIU@DI.UNIPI.IT)

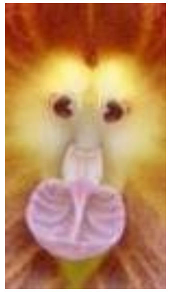
DIPARTIMENTO DI INFORMATICA - UNIVERSITA' DI PISA

IEEE TASK FORCE ON LEARNING FOR STRUCTURED DATA

IN SUPREMÆ DIGNITATIS · 1343 · UNIVERSITÀ DI PISA

www.learning4graphs.org

Why Graphs?

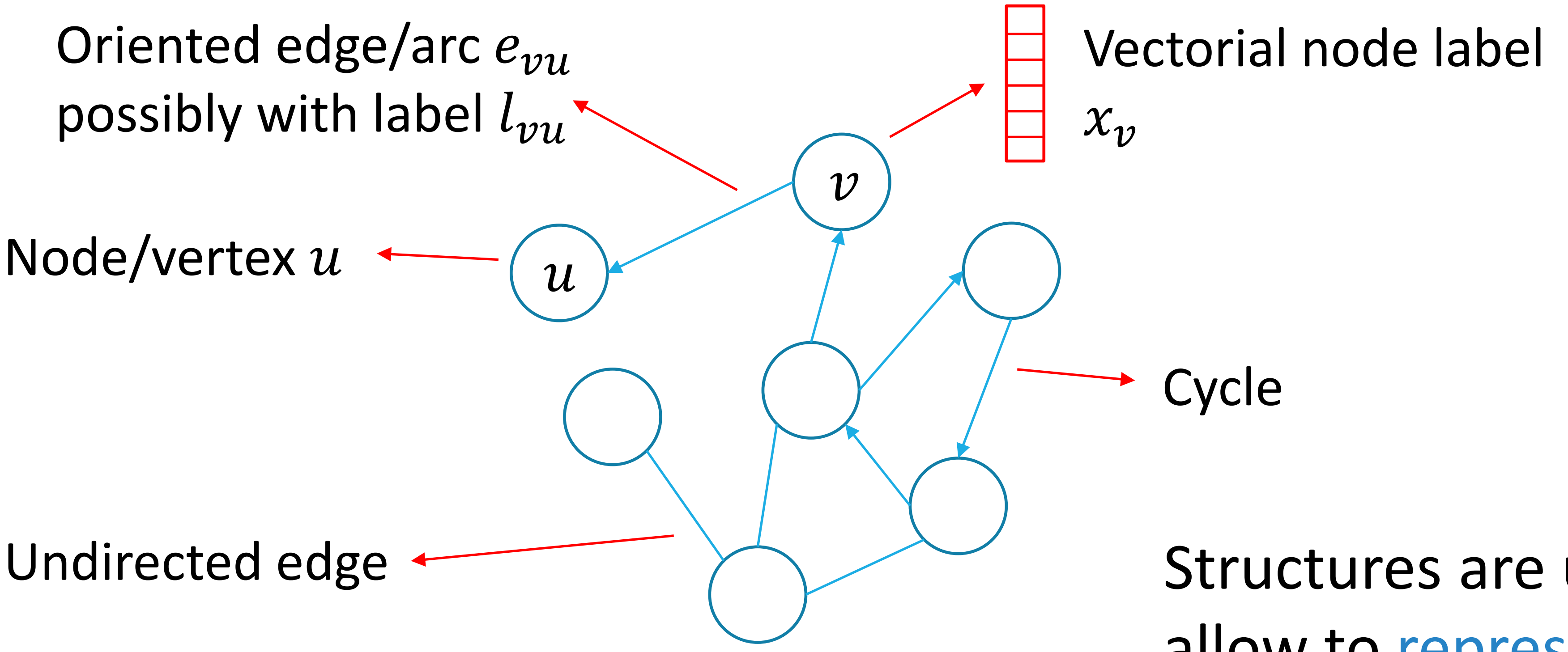


Why Graphs?

Context is fundamental for the correct interpretation of information

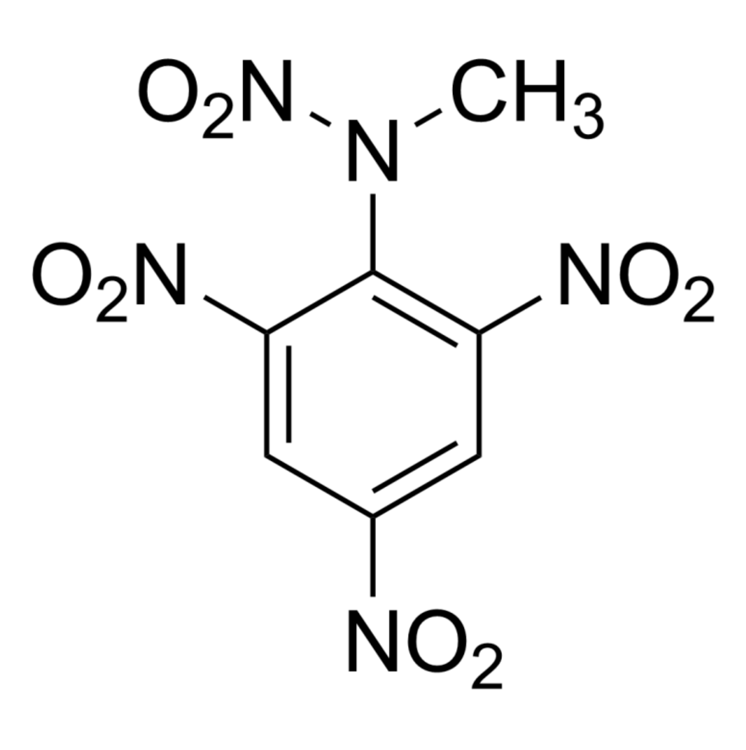
Introduction

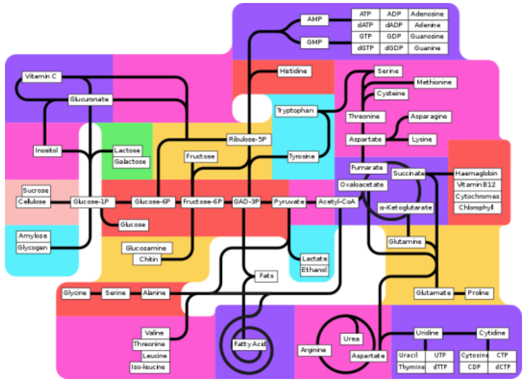
Graph Structured Data

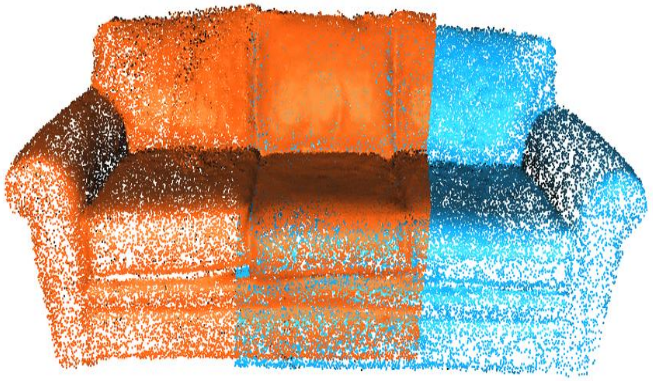


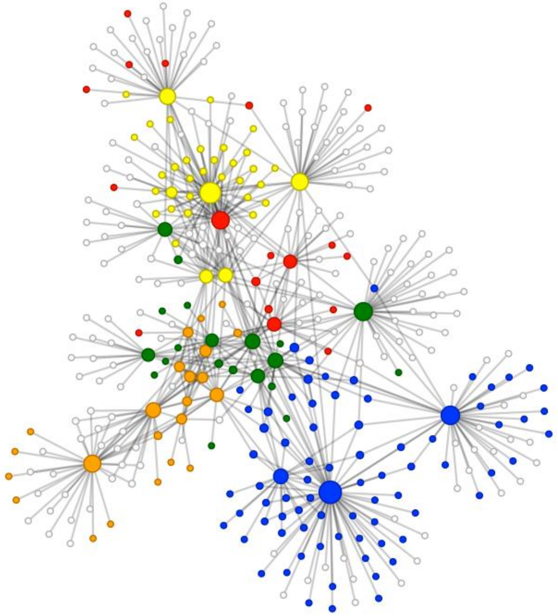
Structures are useful because allow to represent relationships in the data

Motivation – Learning with Graphs

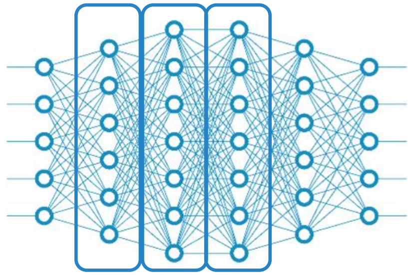




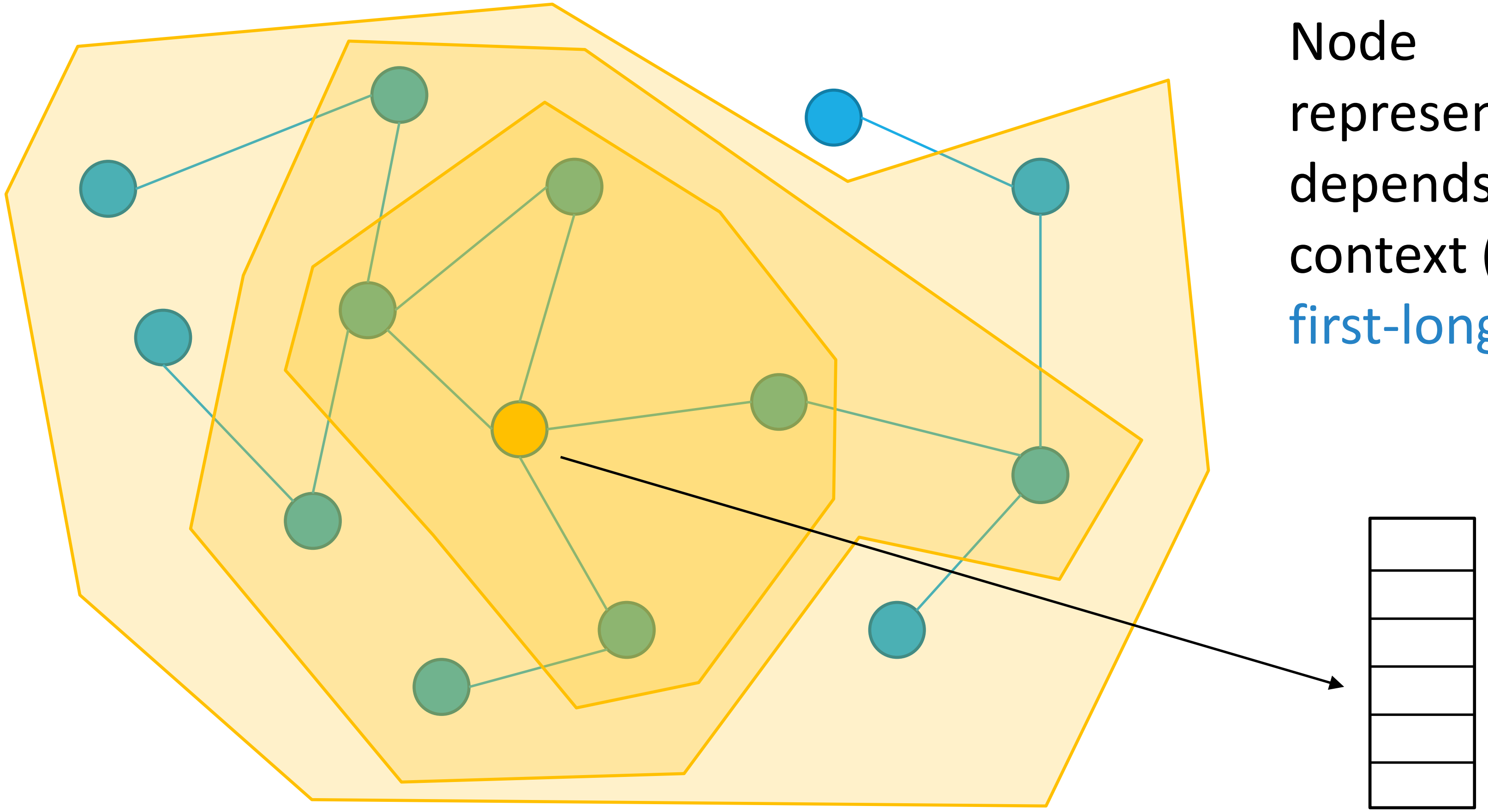




Deep Learning with graphs



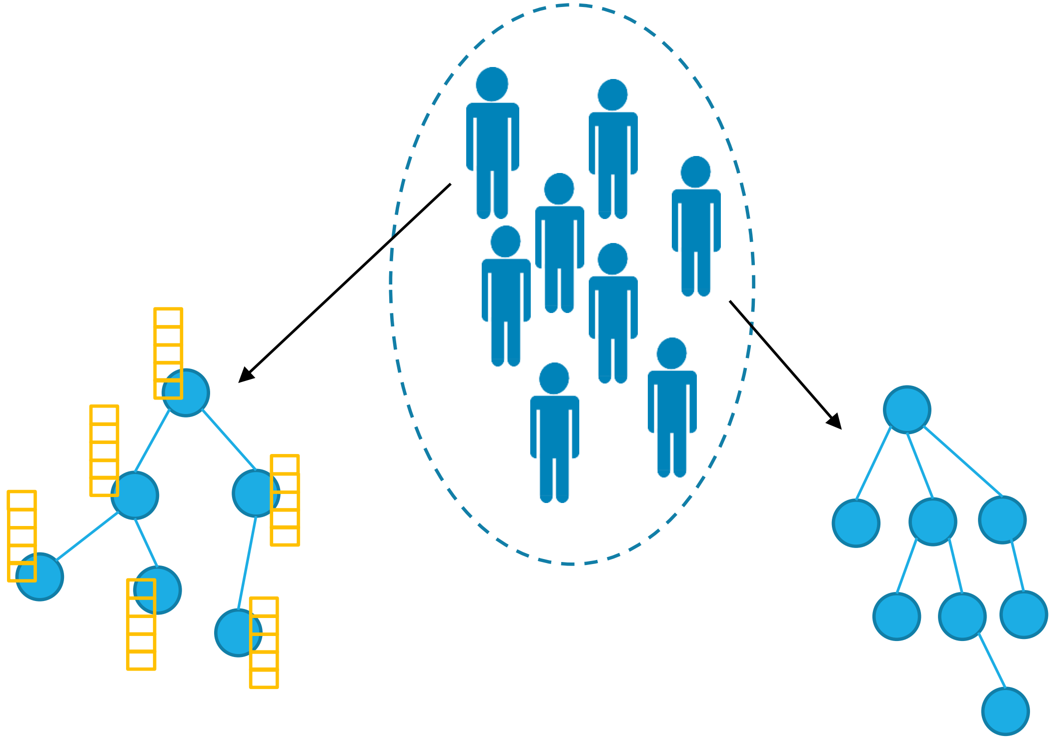
Hierarchical representation learning allows to efficiently diffuse information through graph structure

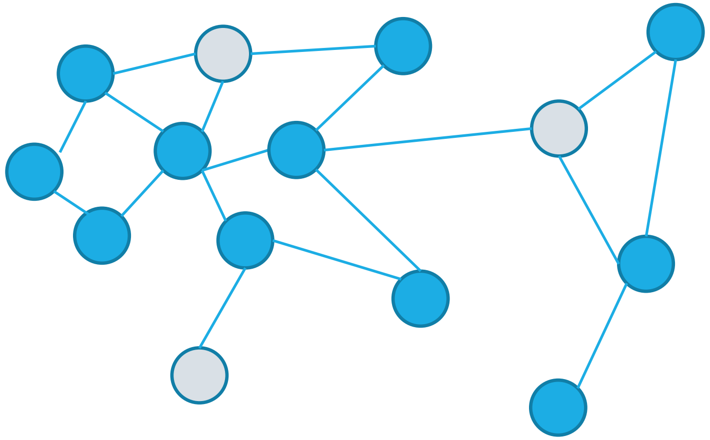


Node representation depends on its context (shorter first-longer later)

Challenges in Learning with Graphs

- Learning from a population where each individual can have different topology and size

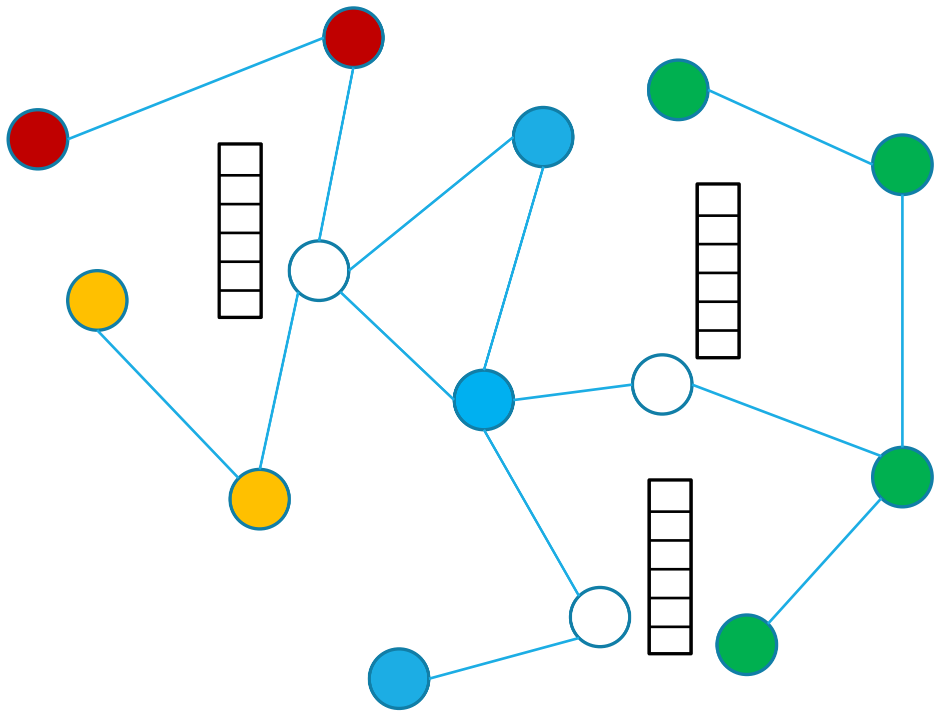




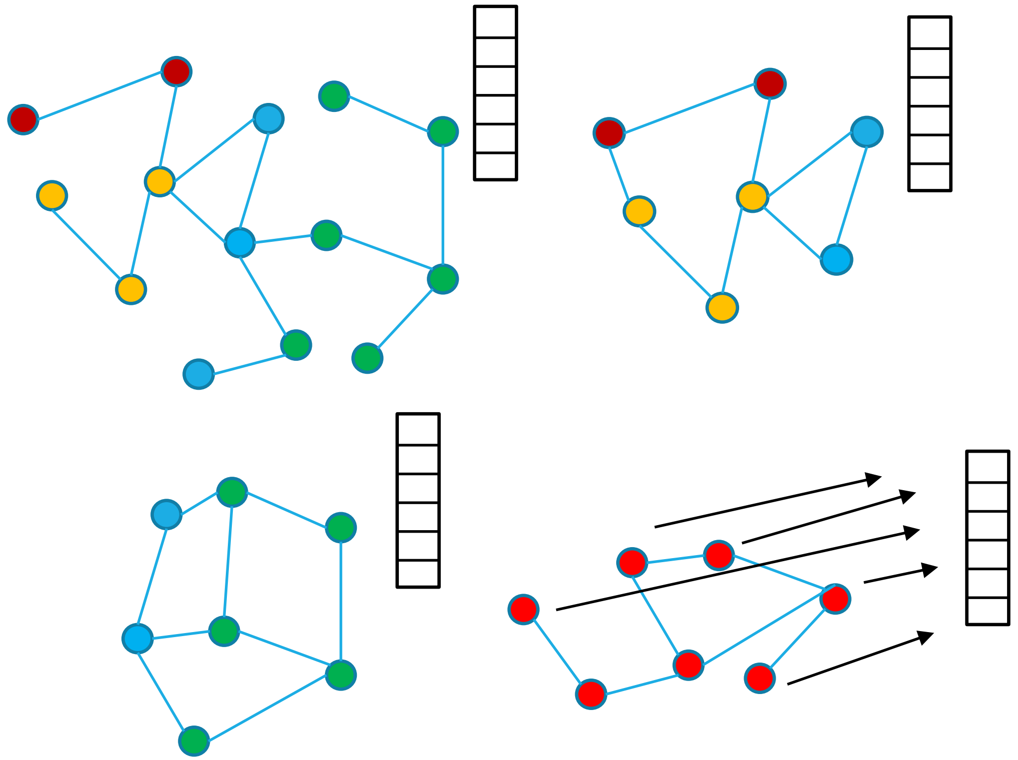
- Dealing with cycles
- Node and edge induction

Predictive Tasks (Transductive & Inductive)

Network data

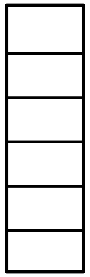


Structure classification/regression

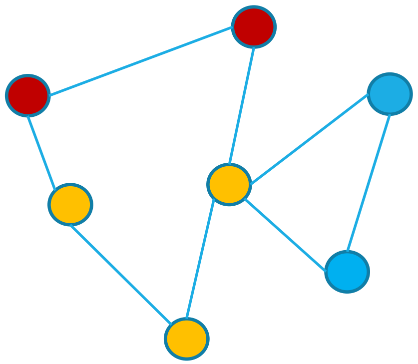


Structure transduction

Given a vectorial and/or structured input

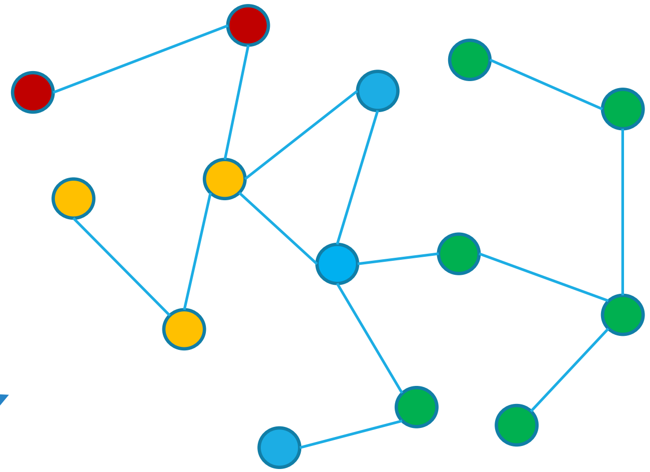


$\boldsymbol{x}$



$$\boldsymbol{y} \sim P(\boldsymbol{y}|\boldsymbol{x})$$

$\boldsymbol{y}$



Learn to generate a structured prediction

Learning with graphs is learning how to deal with cycles

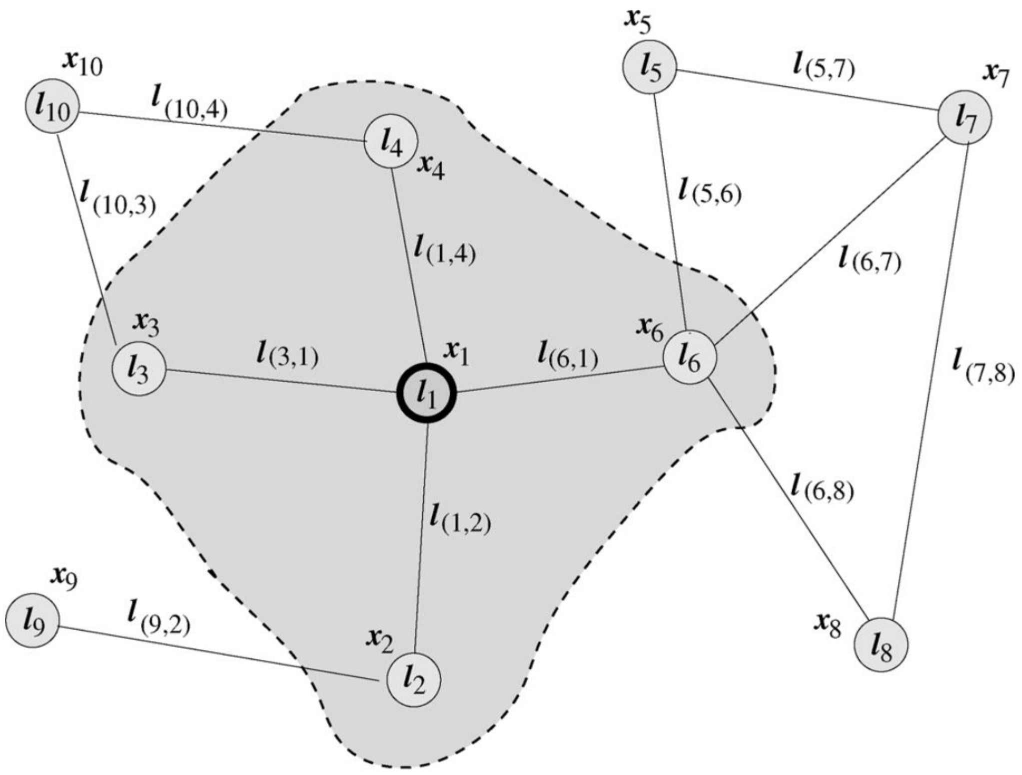
An Historical (and Geographical) Perspective

Early neural network approaches to deal with cyclic graphs of varying topology date back to 2005-2009

IN SUPREMÆ DIGNITATIS 1343
UNIVERSITÀ DI PISA

UNIVERSITÀ DI SIENA
1240

Contractive - Graph Neural Networks (GNN)



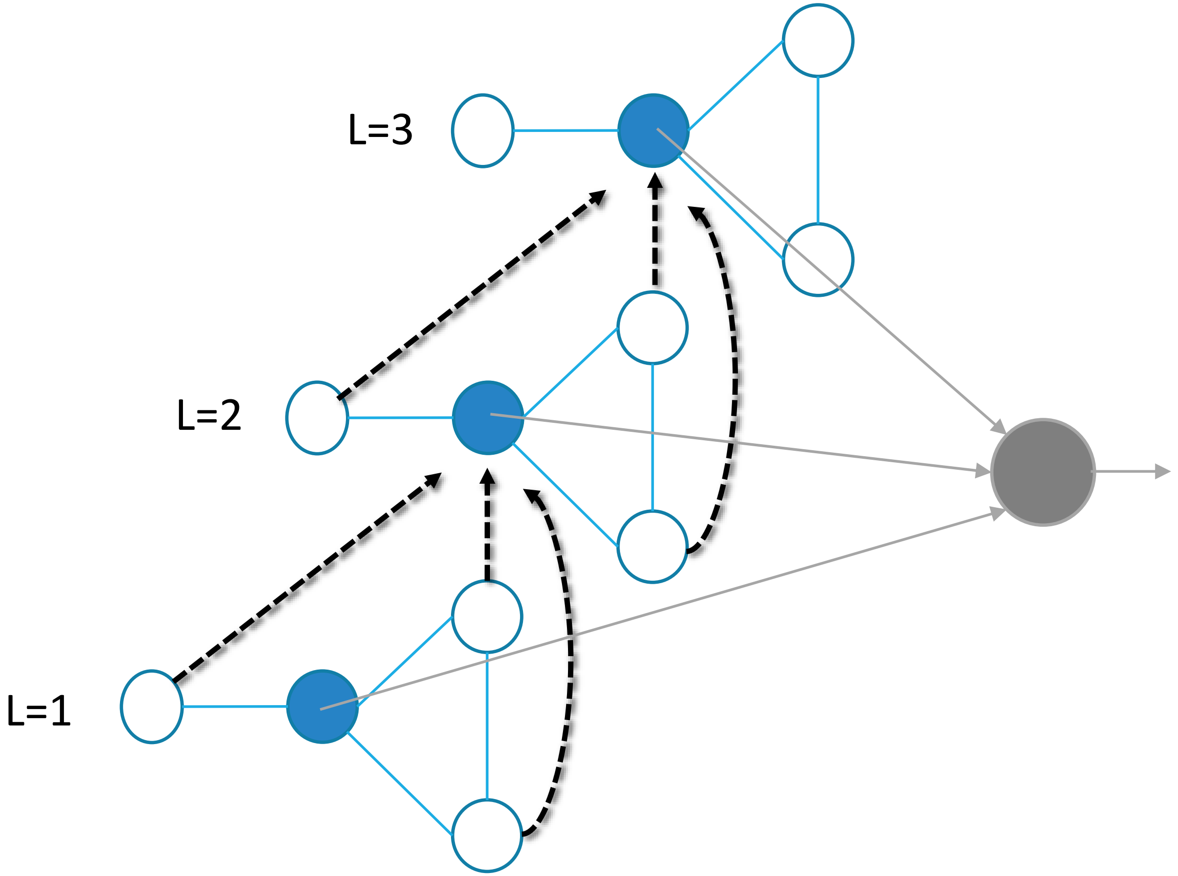
$$x_1 = f_w(l_1, \underbrace{l_{(1,2)}, l_{(3,1)}, l_{(1,4)}, l_{(6,1)}}_{l_{co[1]}}, \underbrace{x_2, x_3, x_4, x_6}_{x_{ne[1]}}, \underbrace{l_2, l_3, l_4, l_6}_{l_{ne[n]}})$$

- Extend the Recurrent/Recursive Neural Network approach to cyclic graphs
- Handle loops through fixed points
- Impose dynamic weight constraints to yield a contractive state mapping

Scarselli et al, TNN 2009

https://sailab.diism.unisi.it/gnn/

Contextual - Neural Networks for Graphs (NN4G)



- A feedforward approach to process graphs
- Handle loops through layering
- Uses context from frozen earlier layers compute the state on the node at current layer
- Layerwise training

A. Micheli, TNN 2009

Deep Graph Networks

A Nomenclature Nightmare

Neural networks for graphs

Deep learning for graphs

CNN for/on graphs

Graph neural networks

Deep Graph Networks

Graph CNN

Learning graph/node embedding

Geometric deep learning

Graph Convolutional Networks

A Survey of Recent Approaches

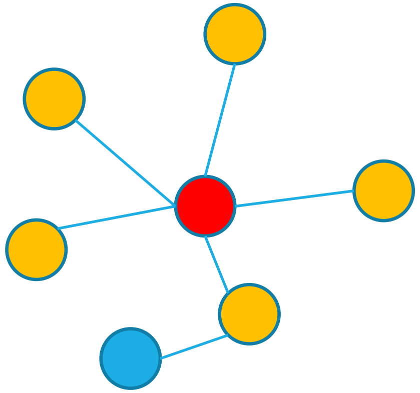
- Convolutional Neural Networks for Graphs
  - Spectral
  - Spatial
- Recurrent Graph Processing
  - Fast graph reservoir networks
- Contextual Graph Processing
  - Neural fingerprints
  - Node embedding, GraphSage
  - GIN, GAT, …

Convolutional Neural Networks for Graphs

How to Perform Convolutions on Graphs

SPATIAL DOMAIN

What is the equivalent of sliding a kernel to aggregate local spatial information?



SPECTRAL DOMAIN

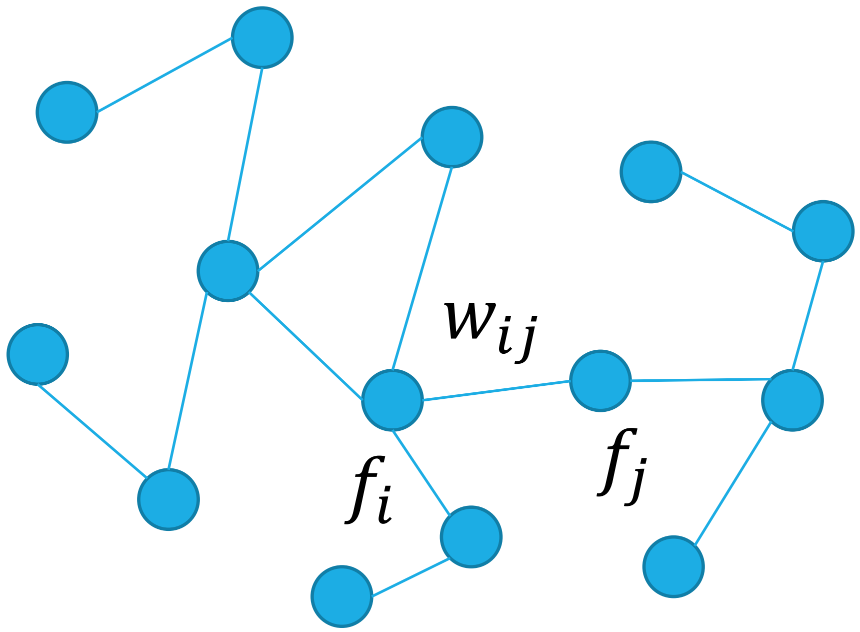
$$\mathcal{F}(f * g) = \mathcal{F}(f) \times \mathcal{F}(g)$$

Exploit the Convolution Theorem and Fourier analysis to perform convolutions in the spectral domain

Decompose a function f as a combination of vectors $\boldsymbol{e}_k$ from an orthonormal basis

Spectral Convolutions

The Spectral Scenario



- Single weighted undirected graph
  - $w_{ij} > 0$ weight of the i-j edge
- Functions f_i attaching values (i.e. labels/signals x_i) to nodes i
- Task: process the signals defined on the graph structure

Spectral Graph Convolution in 1 Slide

Given a graph G, the eigendecomposition of its Laplacian provides an orthonormal basis U which allow to compute the graph convolution of its node signals $\boldsymbol{f}$ with a filter

$$(\boldsymbol{f} *_G \boldsymbol{g}) = \mathcal{F}^{-1}\big(\mathcal{F}(\boldsymbol{f})\,\mathcal{F}(\boldsymbol{g})\big) = U\underbrace{\mathbf{W}(\lambda)}U^T\boldsymbol{f}$$

Convolutional filter **g** in spectral domain

Graph equivalent of the learnable CNN filter matrix **W**

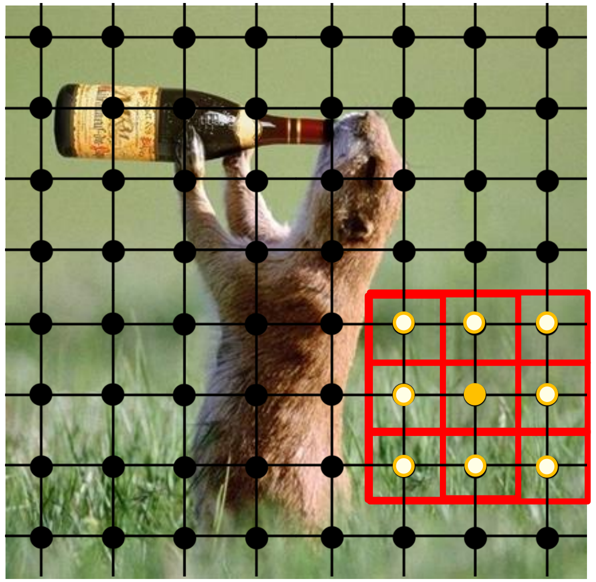
Spectral convolution matrix **W** contains information on the graph Laplacian

Considerations on Spectral Approaches

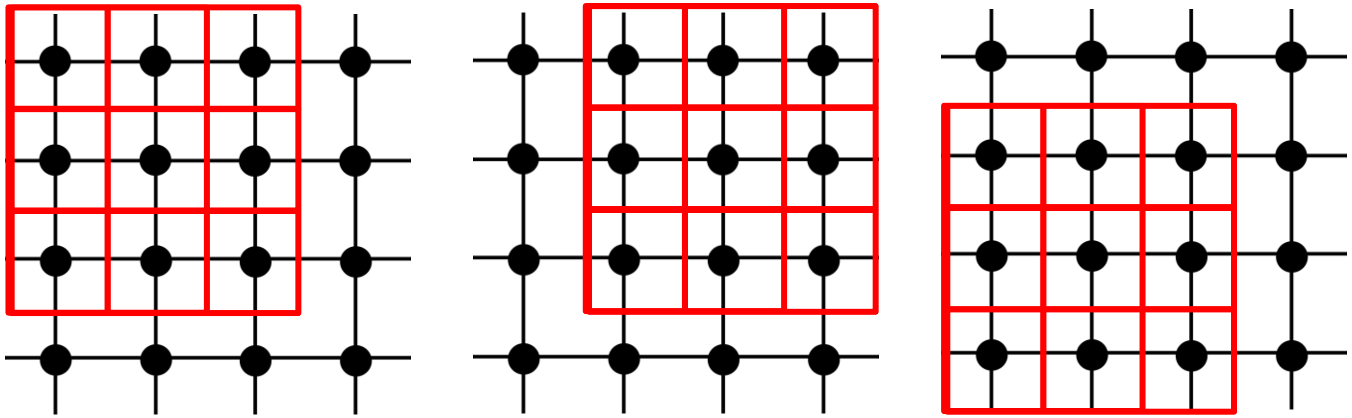
- Cannot handle multiple graphs due to convolution dependency on Laplacian (use on network data tasks)
- Mostly limited to undirected graphs with unlabeled edges
  - Extension to directed graphs using Laplacian block structure and triangular motifs (Benson et al 2016; Monti, Otness, Bronstein 2018)
- Difficult control on context diffusion through the graph structure
- Working with the Laplacian can be impractical for large graphs

Spatial Convolutions

A Graph View on Convolutions



Visual convolutions are graph convolutions on a regular grid

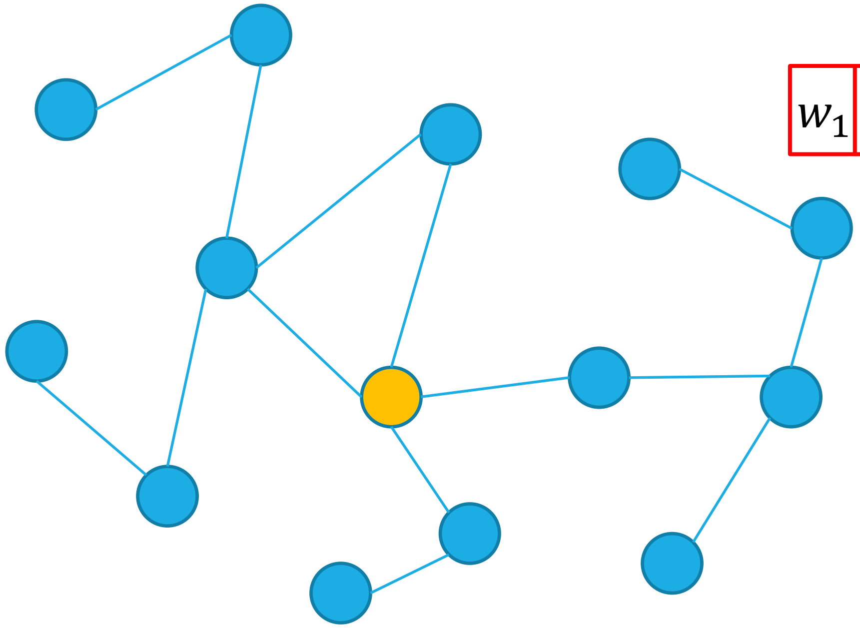


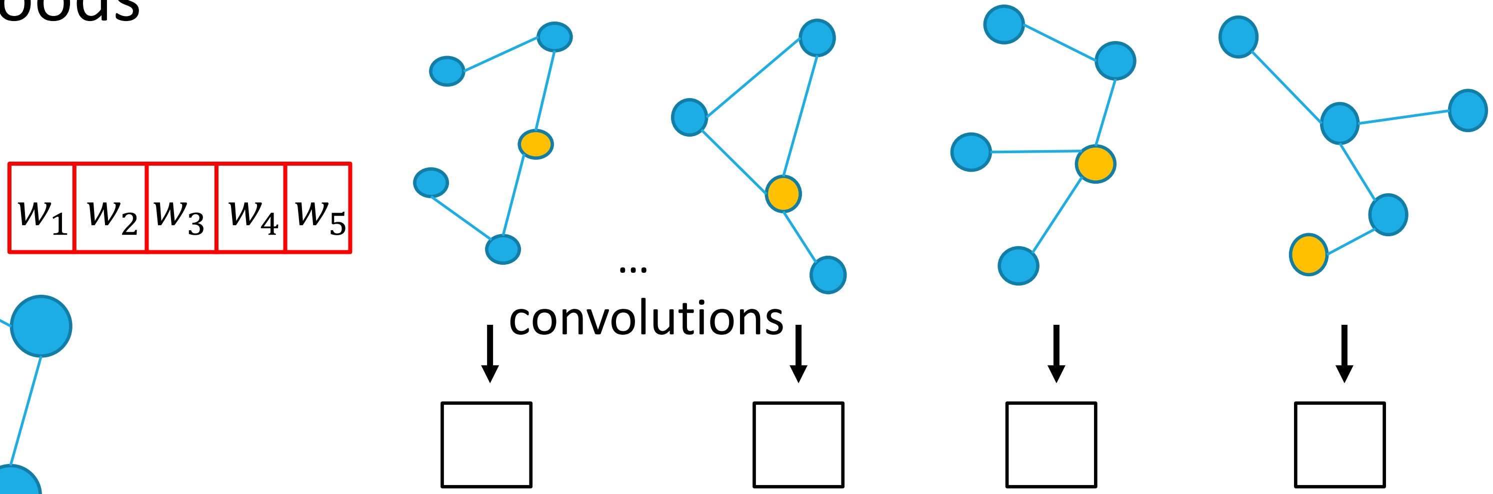
Plus some key assumptions which make it difficult to directly apply them to graphs

- Regular neighborhood
- Existence of a total node ordering

Node Neighborhoods

Example of 4-neighborhoods





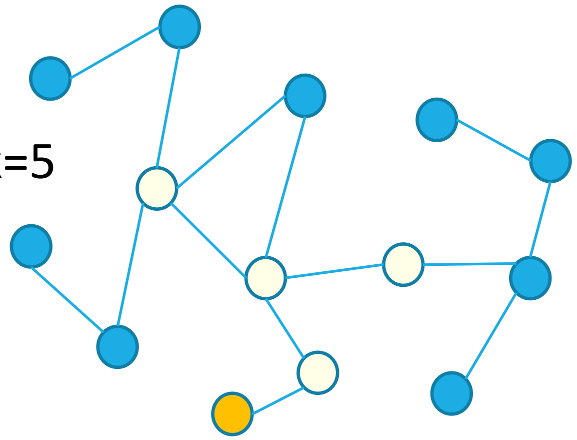
Neighborhoods depend on node ordering: how can I get coherent node ordering across multiple graphs?

PATCHY-SAN

Niepert, Ahmed, Kutzkov, ICML 2016

Leverage graph labelling techniques (e.g. Weisfeiler-Lehman) to determine a coherent ordering within the graph and between the graphs

Neighborhood for k=5



Parametric convolutional filter of size k

w_1	w_2	w_3	w_4	w_5

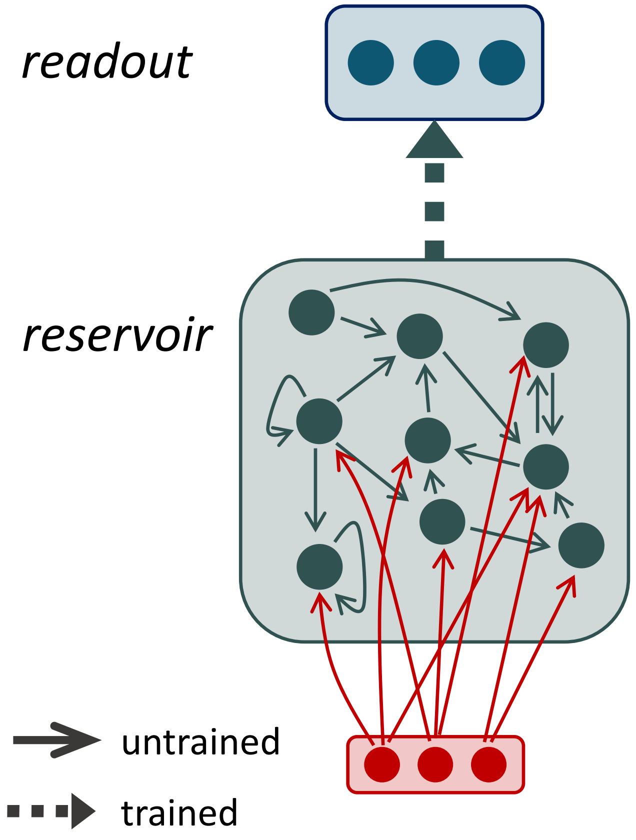
Determining a coherent ordering to match nodes to filter parameters in NP complete (graph normalization)

PATCHY-SAN considerations

- Can handle multiple graphs, undirected and directed, with labels on both edges and nodes
- Can reuse CNN machinery: striding, pooling, …
- Performance relies heavily on quality of the ordering
- Edge labels are used only for computing node ordering
- How to choose neighborhood size?
- Worst case complexity is exponential due to graph normalization

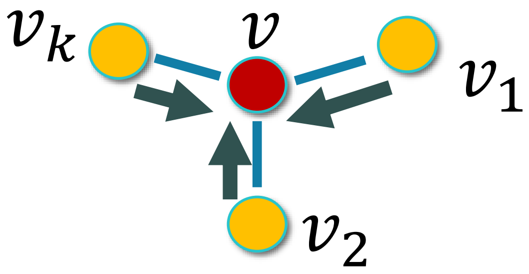
Contractive Approach

Reservoir Computing for Graphs



- Each input graph is encoded by the fixed point of a dynamical system
- The dynamical system is implemented by a hidden layer of recurrent reservoir neurons
- Reservoir Computing (RC)
  - Reservoir neurons do not require learning
  - Only output layer is trained (in closed form)
- Deep Architecture - Multiple levels of reservoir for graphs are stacked to enrich the developed representation

Graph embedding by learning-free neurons



Each vertex in an input graph is encoded by the hidden layer

input weight matrix

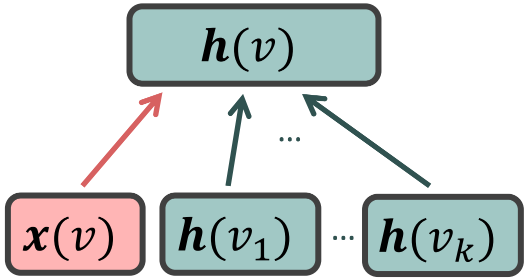
hidden weight matrix

$$\boxed{\mathbf{h}(v)} = \tanh\left(\boxed{\mathbf{V}}\,\boxed{\mathbf{x}(v)} + \sum_{v'\in N(v)} \boxed{\mathbf{W}}\,\boxed{\mathbf{h}(v')}\right)$$

embedding (state) of vertex v

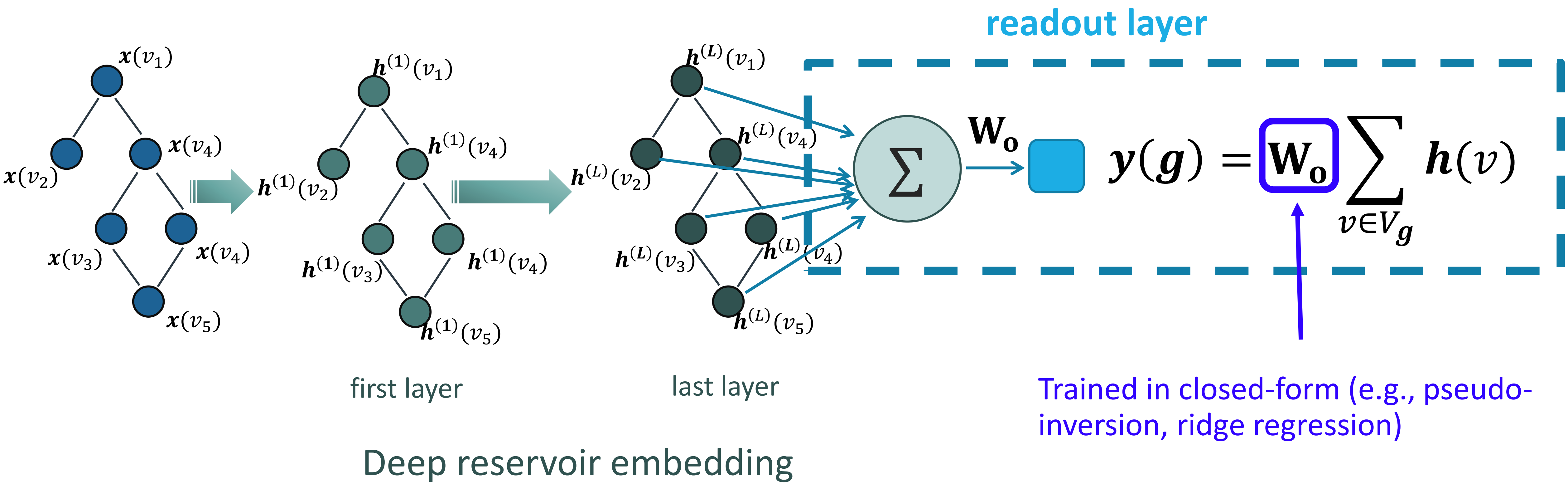
input feature of vertex v

embedding (state) of neighbors of vertex v



Deep Reservoirs for Graphs

Gallicchio & Micheli. *AAAI* 2020.

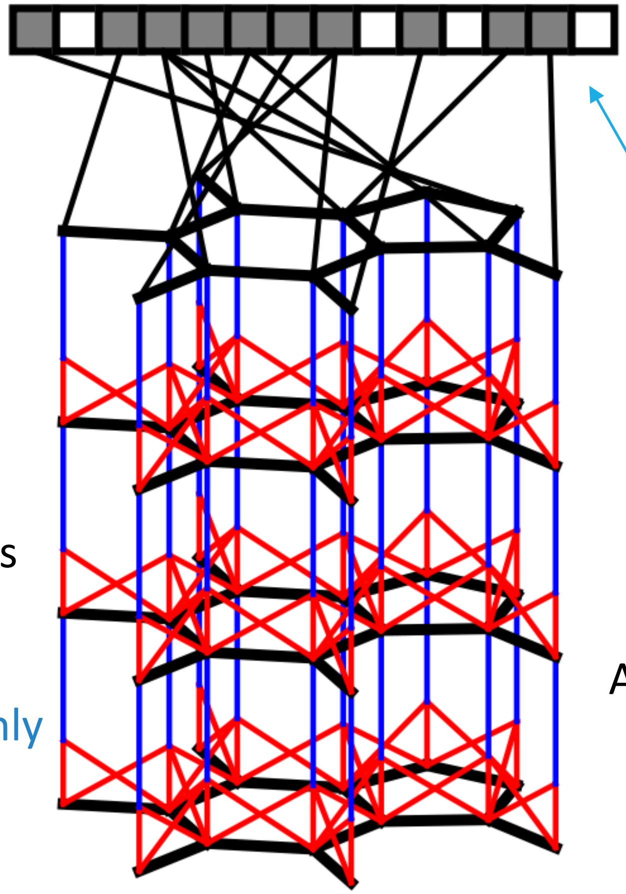


Contextual Graph Processing

Duvenaud, Maclaurin et al, NIPS 2015

Neural (differentiable) Fingerprints

Training by **error backprop**



Differentiable encoding

Current node representation is computed from previous layer contributions only (contextual)

Aggregation

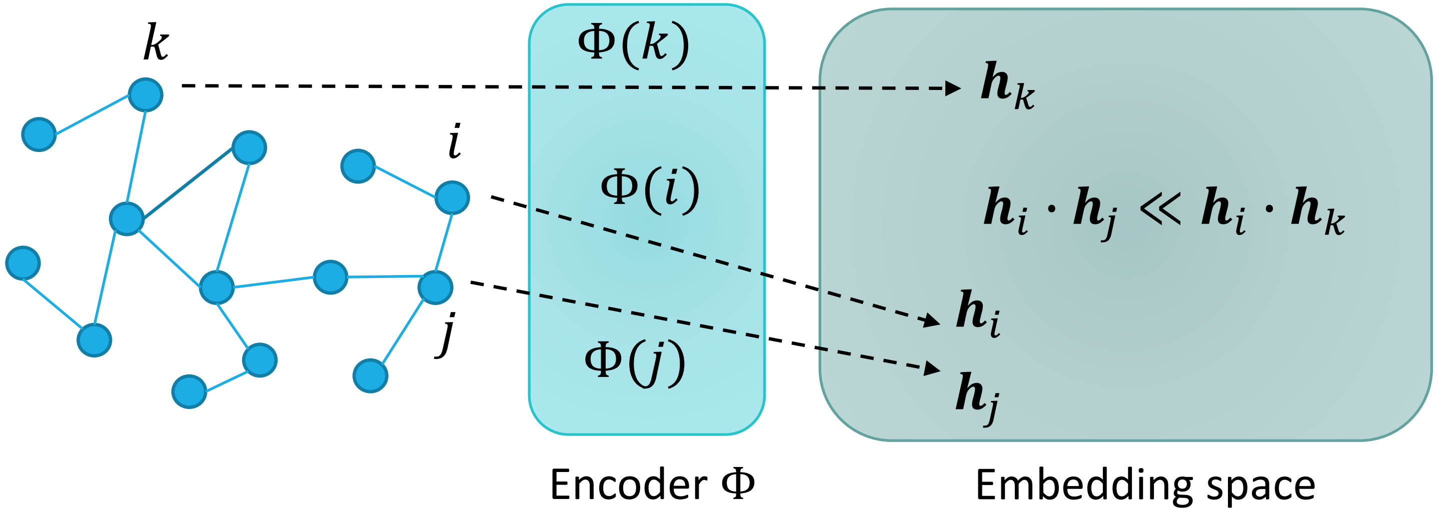
Algorithm 2 Neural graph fingerprints

Input: molecule, radius R, hidden weights $H_1^1 \dots H_R^5$, output weights $W_1 \dots W_R$
Initialize: fingerprint vector $\mathbf{f} \leftarrow \mathbf{0}_S$
for each atom a in molecule
  $\mathbf{r}_a \leftarrow g(a)$ ▷ lookup atom features
for $L = 1$ to R ▷ for each layer
  for each atom a in molecule
    $\mathbf{r}_1 \dots \mathbf{r}_N = \text{neighbors}(a)$
    $\mathbf{v} \leftarrow \mathbf{r}_a + \sum_{i=1}^{N} \mathbf{r}_i$ ▷ sum
    $\mathbf{r}_a \leftarrow \sigma(\mathbf{v} H_L^N)$ ▷ smooth function
    $\mathbf{i} \leftarrow \text{softmax}(\mathbf{r}_a W_L)$ ▷ sparsify
    $\mathbf{f} \leftarrow \mathbf{f} + \mathbf{i}$ ▷ add to fingerprint
Return: real-valued vector $\mathbf{f}$

Node Embedding

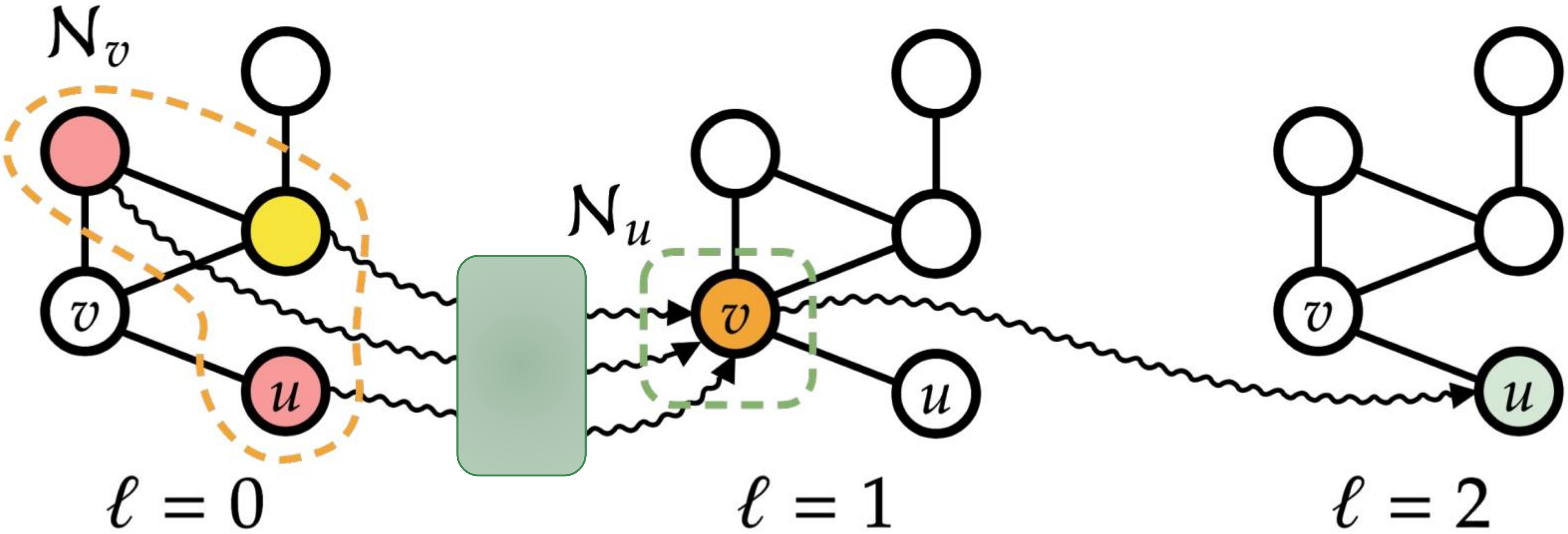
Hamilton, Ying, Leskovec, NIPS 2017

- Encode graph vertices into a vector space where vertex similarities (however defined) are preserved



- Encoding function which can take into account node context when generating the vectorial encoding $\Phi(k) = \Phi(k|G)$ or $\Phi(k|N_k)$

Two Fundamental Principles - Neighborhood Aggregation & Layering



What is inside of the Box?

A learning model of course (e.g. a neural network) including an aggregation function to handle size-varying neighborhoods

A simple model

$$\boldsymbol{h}_v^{\mathrm{l}} = \sigma\left(\boldsymbol{W}_l AGG\left(\left\{\boldsymbol{h}_i^{l-1} : i \in N(v)\right\}\right), \widehat{\boldsymbol{W}}_l \boldsymbol{h}_v^{l-1}\right)$$

The graph convolutional layer

$$\underbrace{\mathbf{h}_v^{\ell+1}}_{\text{state}} = \overbrace{\phi^{\ell+1}}^{\text{MLP/Linear}}\Big(\mathbf{h}_v^{\ell},\ \underbrace{\Psi}_{\text{perm. invariant function}}(\{\overbrace{\psi^{\ell+1}}^{\text{MLP/Linear}}(\mathbf{h}_u^{\ell}) \mid u \in \mathcal{N}_v\})\Big)$$

Variants/extensions:

- **Edge-aware convolution**
- **Attention over neighbours**
- **Laplacian-normalized**

Model	Neighborhood Aggregation $\mathbf{h}_v^{\ell+1}$
NN4G [88]	$\sigma\Big(\mathbf{w}^{\ell+1^T}\mathbf{x}_v + \sum_{i=0}^{\ell}\sum_{c_k\in\mathcal{C}}\sum_{u\in\mathcal{N}_v^{c_k}} w_{c_k}^i * \mathbf{h}_u^i\Big)$
GNN [104]	$\sum_{u\in\mathcal{N}_v} MLP^{\ell+1}\Big(\mathbf{x}_u, \mathbf{x}_v, \mathbf{a}_{uv}, \mathbf{h}_u^{\ell}\Big)$
GraphESN [44]	$\sigma\Big(\mathbf{W}^{\ell+1}\mathbf{x}_u + \hat{\mathbf{W}}^{\ell+1}[\mathbf{h}_{u_1}^{\ell}, \dots, \mathbf{h}_{u_{\mathcal{N}_v}}^{\ell}]\Big)$
GCN [72]	$\sigma\Big(\mathbf{W}^{\ell+1}\sum_{u\in\mathcal{N}(v)}\mathbf{L}_{vu}\mathbf{h}_u^{\ell}\Big)$
GAT [120]	$\sigma\Big(\sum_{u\in\mathcal{N}_v}\alpha_{uv}^{\ell+1} * \mathbf{W}^{\ell+1}\mathbf{h}_u\Big)$
ECC [111]	$\sigma\Big(\frac{1}{\lvert\mathcal{N}_v\rvert}\sum_{u\in\mathcal{N}_v} MLP^{\ell+1}(\mathbf{a}_{uv})^T\mathbf{h}_u^{\ell}\Big)$
R-GCN [105]	$\sigma\Big(\sum_{c_k\in\mathcal{C}}\sum_{u\in\mathcal{N}_v^{c_k}}\frac{1}{\lvert\mathcal{N}_v^{c_k}\rvert}\mathbf{W}_{c_k}^{\ell+1}\mathbf{h}_u^{\ell} + \mathbf{W}^{\ell+1}\mathbf{h}_v^{\ell}\Big)$
GraphSAGE [54]	$\sigma\Big(\mathbf{W}^{\ell+1}(\frac{1}{\lvert\mathcal{N}_v\rvert}[\mathbf{h}_v^{\ell}, \sum_{u\in\mathcal{N}_v}\mathbf{h}_u^{\ell}])\Big)$
CGMM [3]	$\sum_{i=0}^{\ell} w^i * \Big(\sum_{c_k\in\mathcal{C}} w_{c_k}^i * (\frac{1}{\lvert\mathcal{N}_v^{c_k}\rvert}\sum_{u\in\mathcal{N}_v^{c_k}}\mathbf{h}_u^i)\Big)$
GIN [131]	$MLP^{\ell+1}\Big((1+\epsilon^{\ell+1})\mathbf{h}_v^{\ell} + \sum_{u\in\mathcal{N}_v}\mathbf{h}_u^{\ell}\Big)$

Graph Isomorphism Network (a.k.a. sum is better)

Xu et al, ICLR 2019

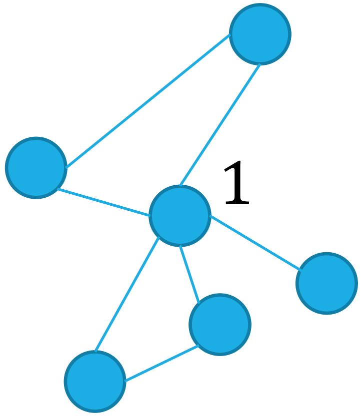
- A study of GNN expressivity w.r.t. WL test of graph isomorphism
- Choice of aggregation functions influences what structures can be recognized
- Propose a simple aggregation and concatenation model

$$h_v^{(k)} = \mathrm{MLP}^{(k)}\left(\left(1+\epsilon^{(k)}\right)\cdot h_v^{(k-1)} + \sum_{u\in\mathcal{N}(v)} h_u^{(k-1)}\right)$$

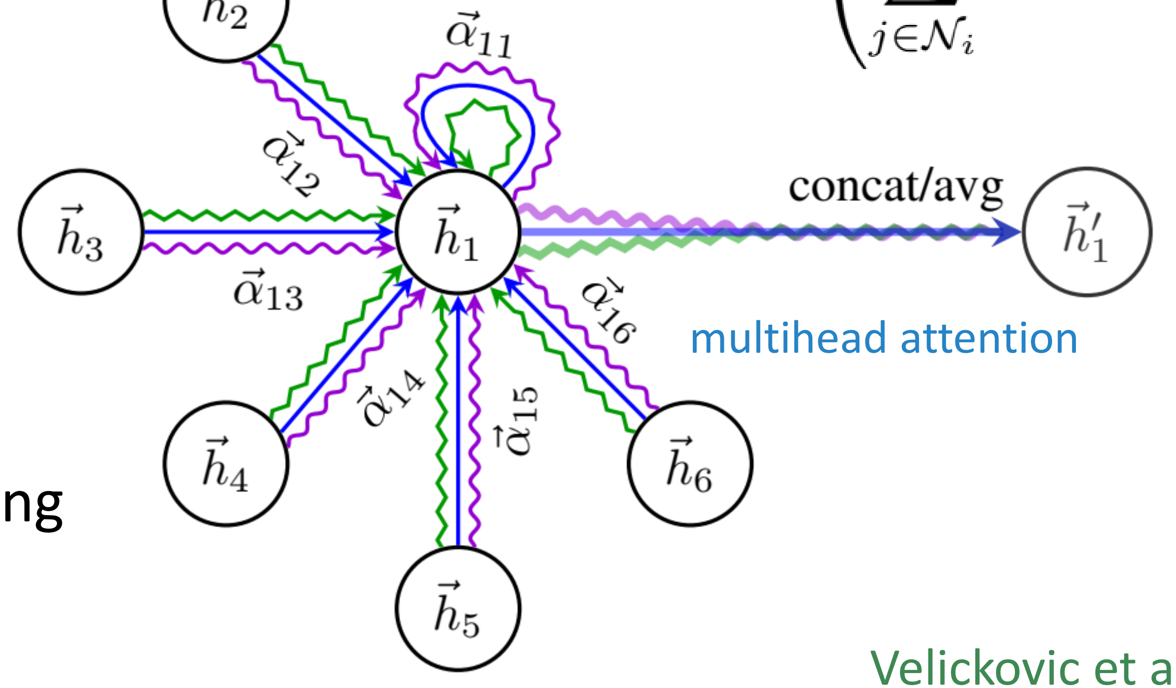
Basically the NN4G approach

$$h_G = \mathrm{CONCAT}\left(\mathrm{READOUT}\left(\left\{h_v^{(k)} \middle| v\in G\right\}\right)\middle| k = 0,1,\cdots,K\right)$$

Graph Attention



$$\vec{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} \vec{h}_j\right)$$



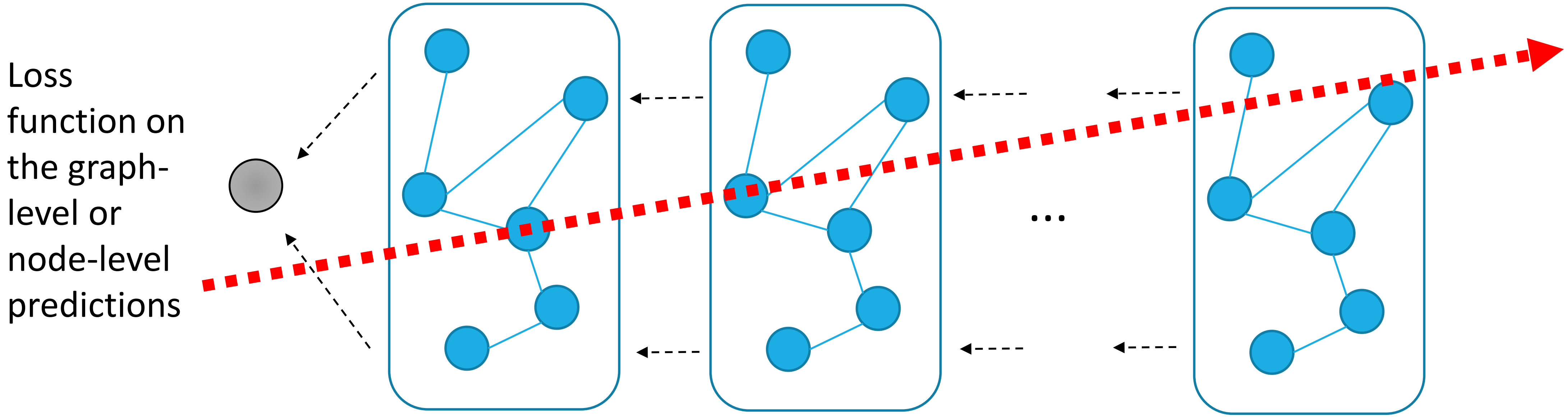
multihead attention

Learning to weight contribution of other nodes when aggregating to form the node embedding

Velickovic et al, ICLR 2018

Training the Embedding

Backpropagate from the (graph or node level) error computed from the top layer embeddings to the early layers



Can also be unsupervisedly trained by using structure induced notions of node similarity (e.g. Node2Vec)

End-to-end Contextual Processing Recap

- Exploit contextual approach to avoid complex neighborhood construction strategies
- No causal dependencies within layers, hence need no fixed-point recurrence
- Restrict context to the preceding layer alone (less general than NN4G)
- Number of layers is typically small (computational, parameterization and oversmoothing issues related to end-to-end training)
- Embedding are either task-dependent (supervised learning) or need to hand-define similarity in node space (unsupervised learning)

Computational Intelligence and
Machine Learning Group

Deep Graph Networks – Generative approaches & Research Directions

DAVIDE BACCIU (DAVIDE.BACCIU@DI.UNIPI.IT)

DIPARTIMENTO DI INFORMATICA - UNIVERSITA' DI PISA

IEEE TASK FORCE ON LEARNING FOR STRUCTURED DATA

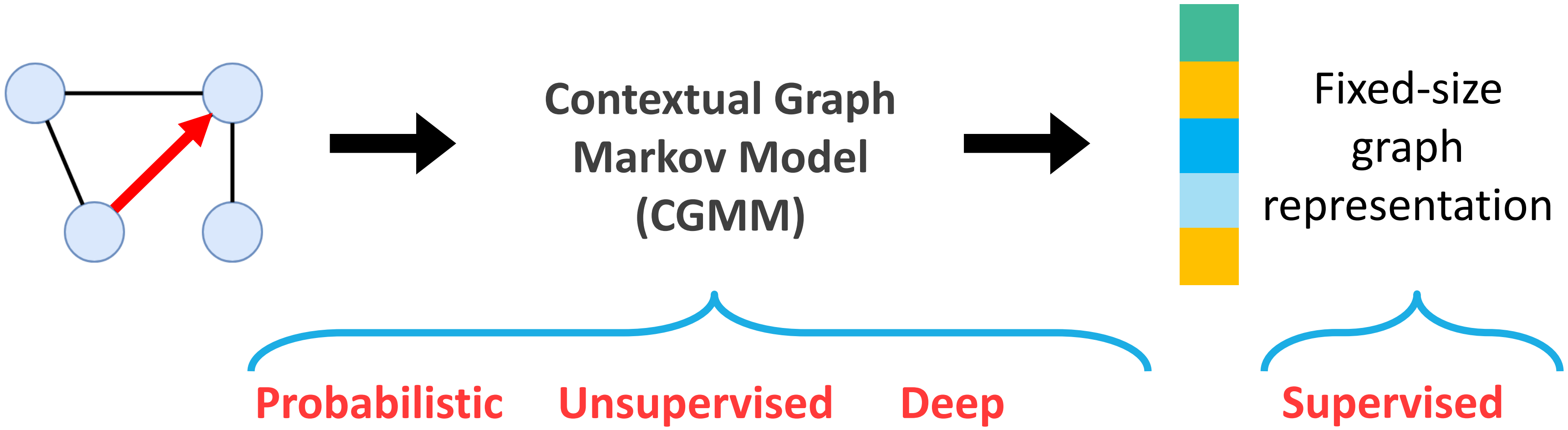
www.learning4graphs.org

Unsupervised Structure Embeddings

… WITH A PROBABILISTIC TWIST

Generative learning for graphs

- General, efficient and scalable architecture
- Handle arbitrary structure (directed, undirected or mixed), labelled edges and nodes
- Learn in both supervised and unsupervised way

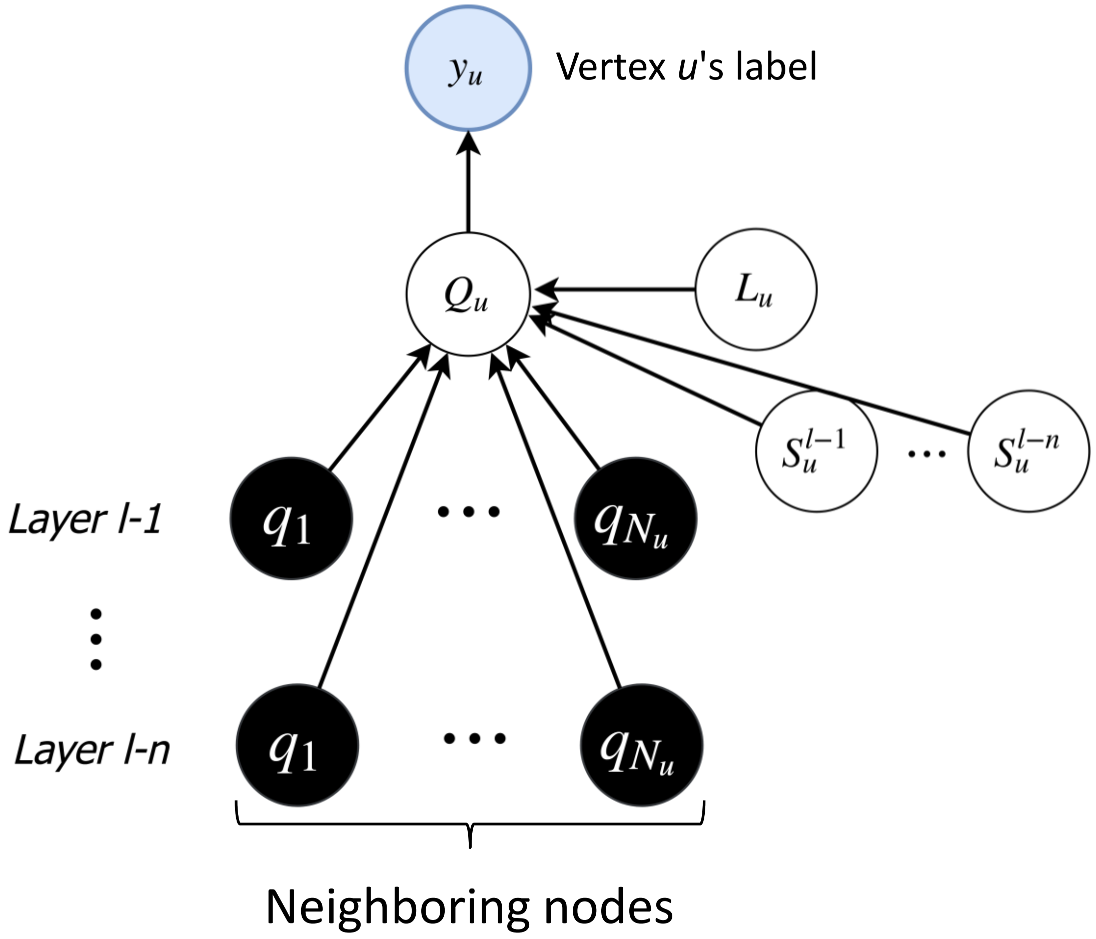


Bacciu, Errica, Micheli , ICML 2018

CGMM in a nutshell

The single layer graphical model

- **Extension** of a standard mixture model
  - Discrete variables
- Likelihood becomes **intractable**
  - Exploit a **Switching Parent approximation**
- Consider only the **direct neighborhood**
- **Full** stationarity



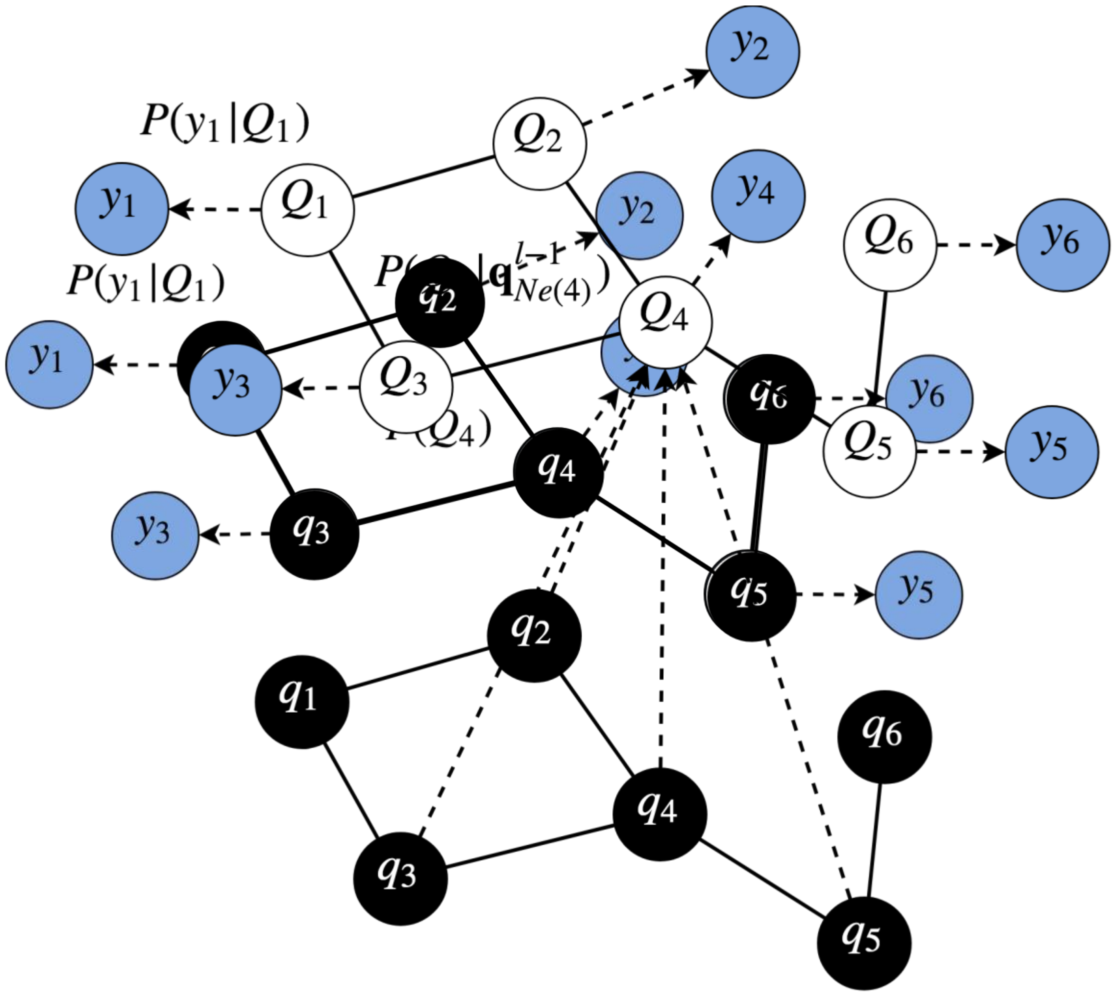
How to build the model

1. Map the graph to the model (base case)

2. Perform inference
   and freeze states

3. Add a new layer
   and use frozen states
   as observed variables
   in the graphical model

Go back to step 2



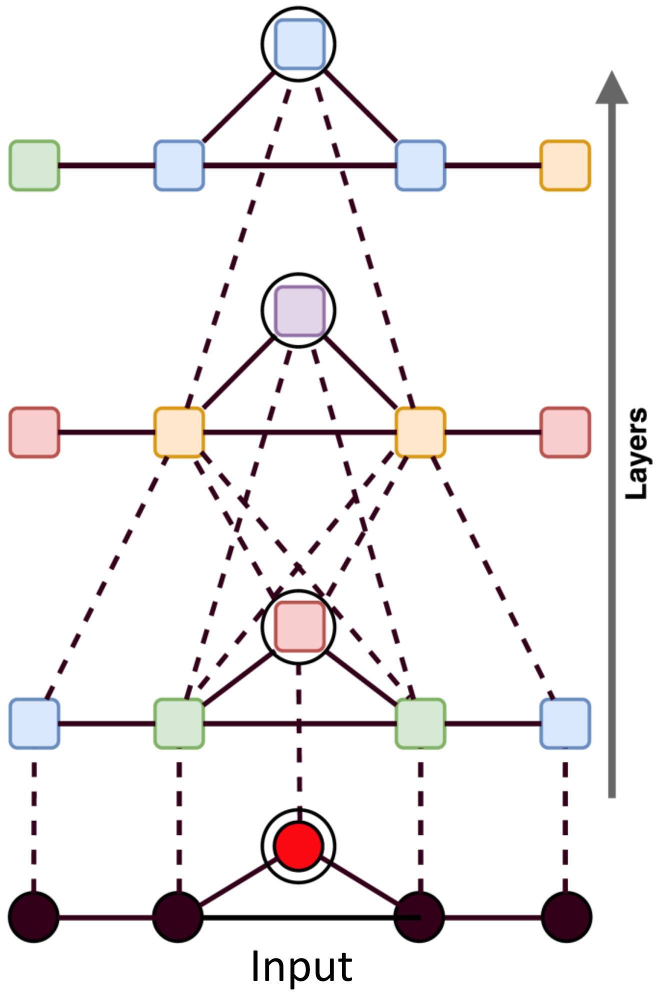
CGMM in a nutshell

Symmetric context spreading between vertexes

- Each layer is trained in **isolation**
- Inference computes hidden states' assignments

  Variables encode information
- The **architecture** diffuses information

  Deeper net → Wider context window



Learning phase

A maximum likelihood approach to learning

$$\mathcal{L} = \prod_{g \in G} \prod_{u \in g} \sum_{i}^{C} P^{l}(y_u | Q_u = i) P^{l}(Q_u = i | \underbrace{\mathbf{q}^{\mathbf{L_{prec}}}_{\mathcal{N}(\mathbf{u})}}(\mathbf{g})))$$

Split by layer

and by arc

$$= \prod_{g \in G} \prod_{u \in g} \sum_{i}^{C} P^{l}(y_u | Q_u = i) \sum_{\bar{l} \in \bar{L}} P(L_u = \bar{l}) \sum_{a=1}^{A} P^{\bar{l}}(S_u = a) \underbrace{\frac{\sum_{v \in \mathcal{N}^a(u)} P^{\bar{l},a}(Q_u = i | q_v)}{|\mathcal{N}^a(u)|}}_{\text{Average of the remaining contributions}}$$

Trained by EM

Assumption: i.i.d. graphs

- Emission distrib.
- Switching Parents distrib.
- Transition distrib.

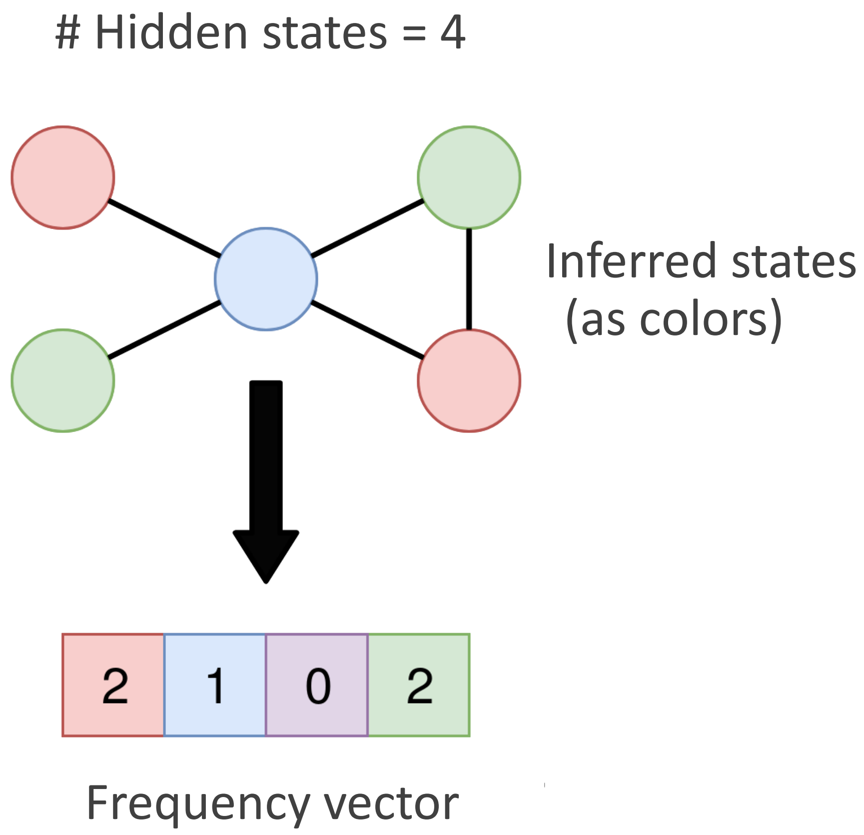
Inference

- Finding the most likely **state assignment**

$$\max_i P(y_u|Q_u = i)P(Q_u = i|\mathbf{q}_{\mathcal{N}(u)})$$

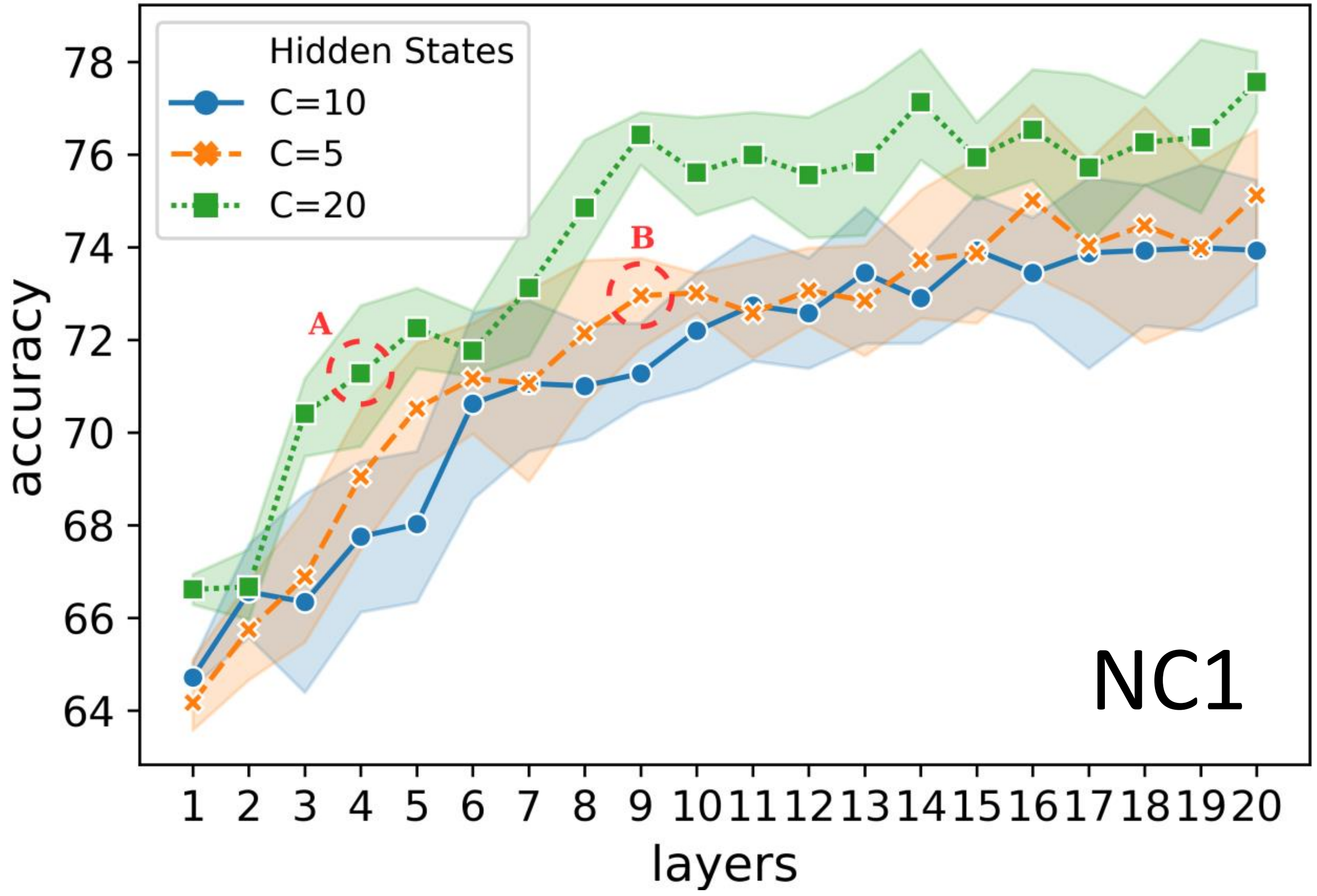
- The inferred latent states are used as observable variables in subsequent layers

- A **fixed-size vector of states frequencies** as graph encoding



CGMM – Depth Matters…

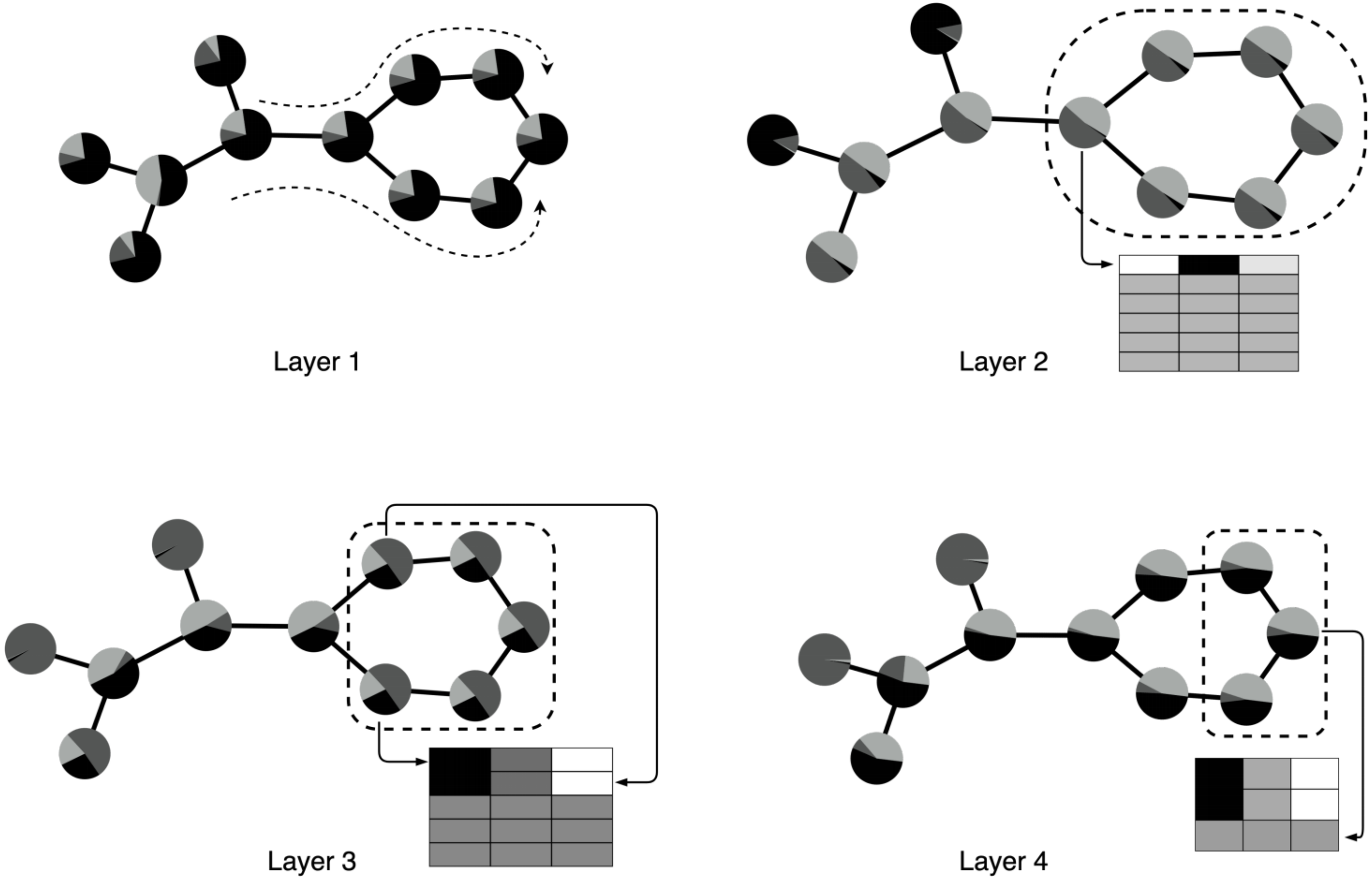
…possibly more than width



Bacciu, Errica, Micheli , JMLR 2020

Interpreting CGMM

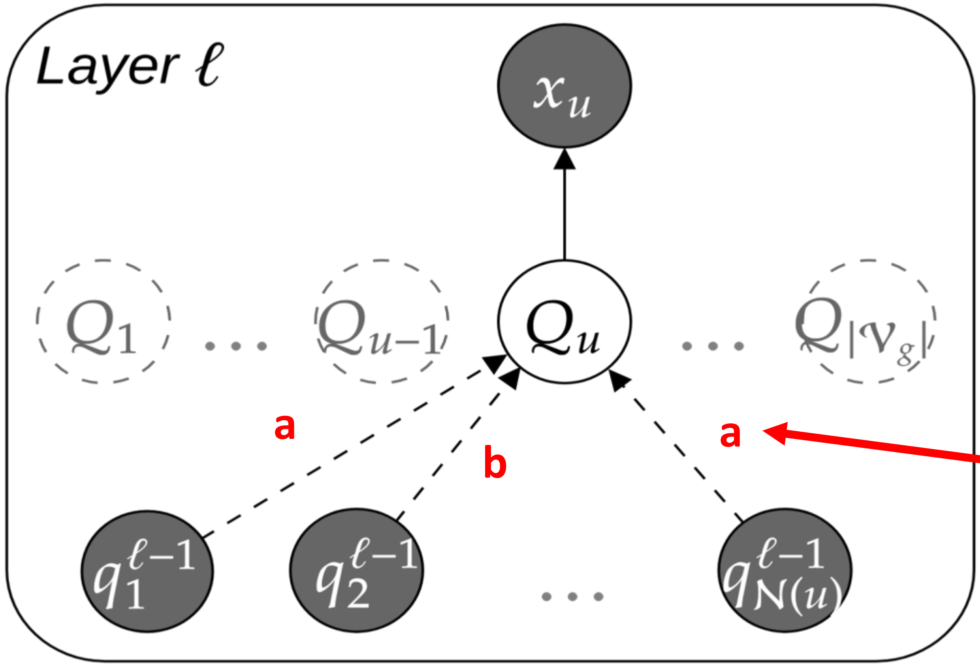
Thanks to the probabilistic approach

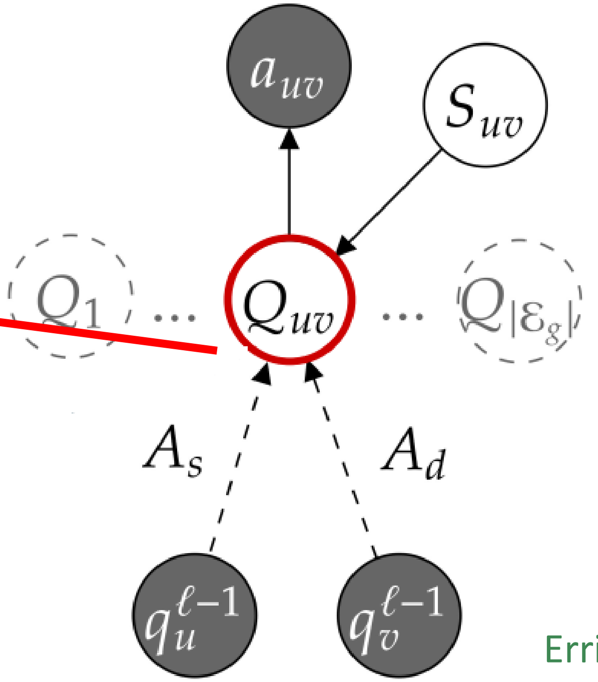


Bacciu, Errica, Micheli , JMLR 2020

What about edge labels?

...a bit less dumbly if you add a second edge encoding module

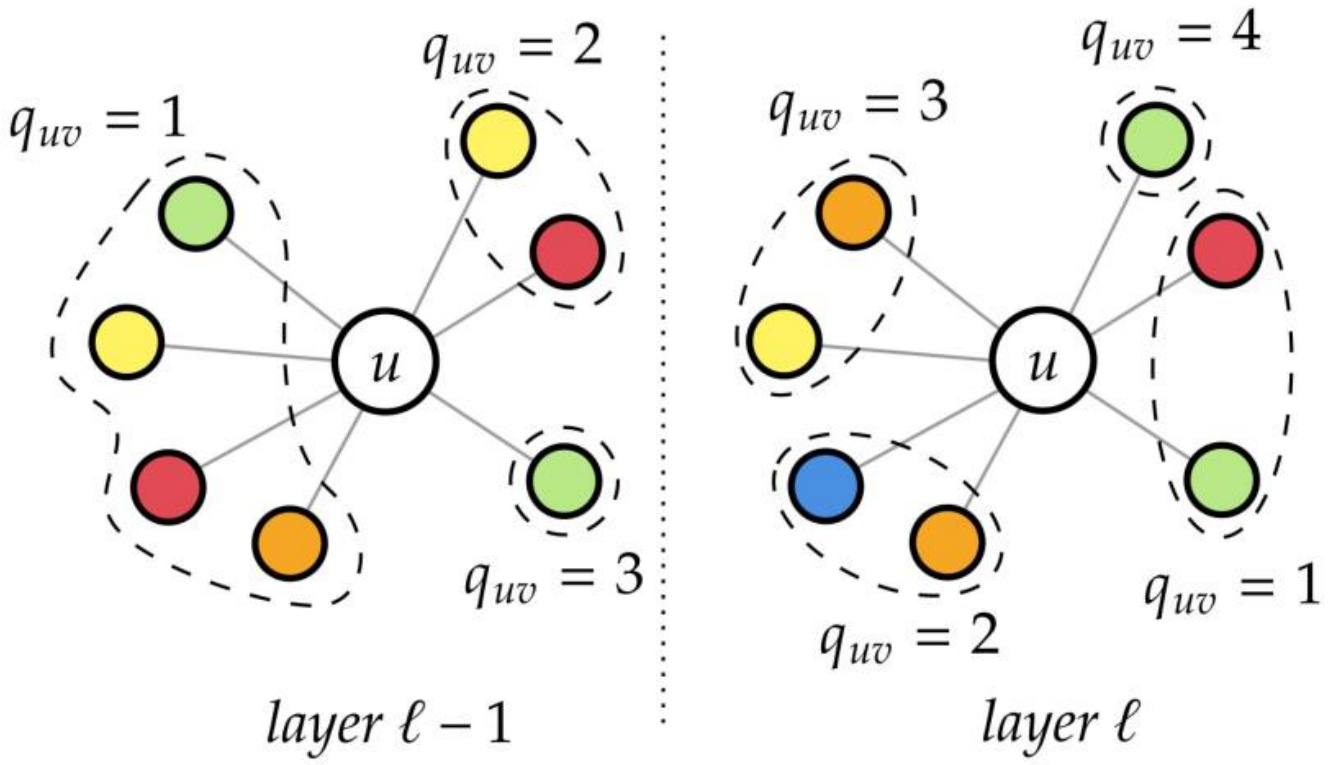




Dumbly, for discrete edge labels...

Errica et al IJCNN 2021

You also earn some interesting complimentary perks



- ✓ Works well also when edge labels are not available
- ✓ Dynamic neighborhood aggregation
- ✓ Provides richer node/graph embeddings

Deep Graph Infomax

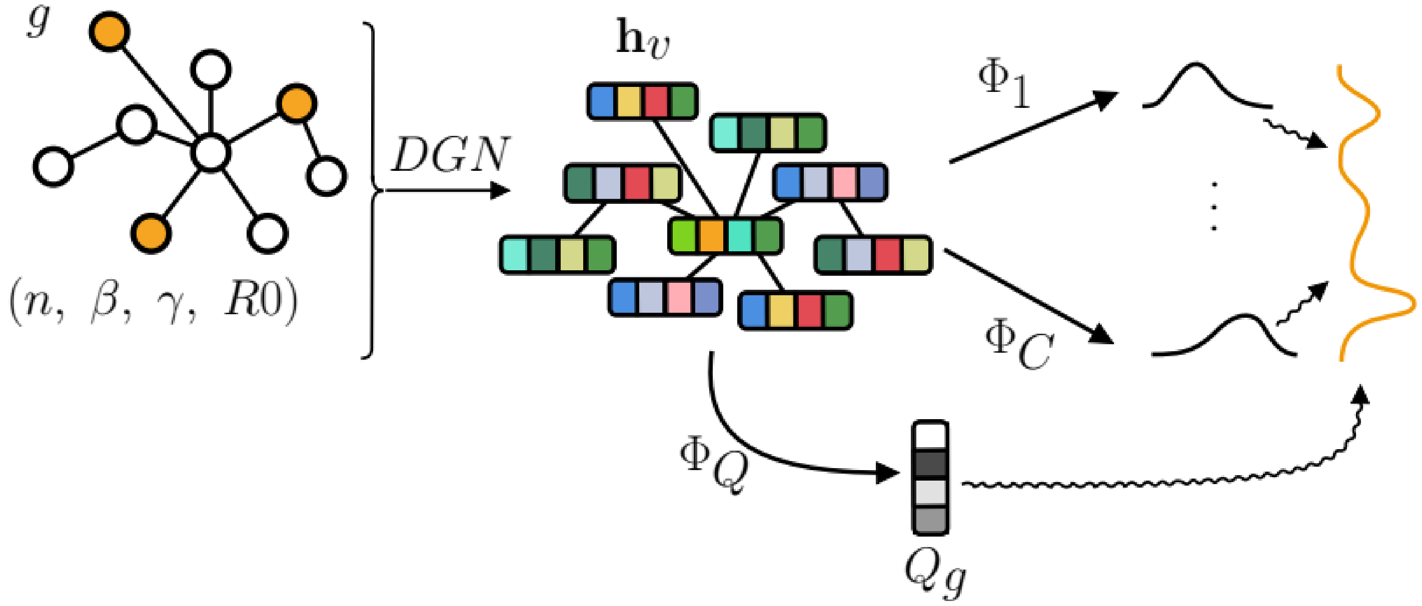
Unsupervised approach seeking node representations that capture the global information content of the entire graph

$$\mathcal{L} = \frac{1}{N+M}\left(\sum_{i=1}^{N}\mathbb{E}_{(\mathbf{X},\mathbf{A})}\left[\log \mathcal{D}\left(\vec{h}_i, \vec{s}\right)\right] + \sum_{j=1}^{M}\mathbb{E}_{(\widetilde{\mathbf{X}},\widetilde{\mathbf{A}})}\left[\log\left(1-\mathcal{D}\left(\vec{\widetilde{h}}_j, \vec{s}\right)\right)\right]\right)$$

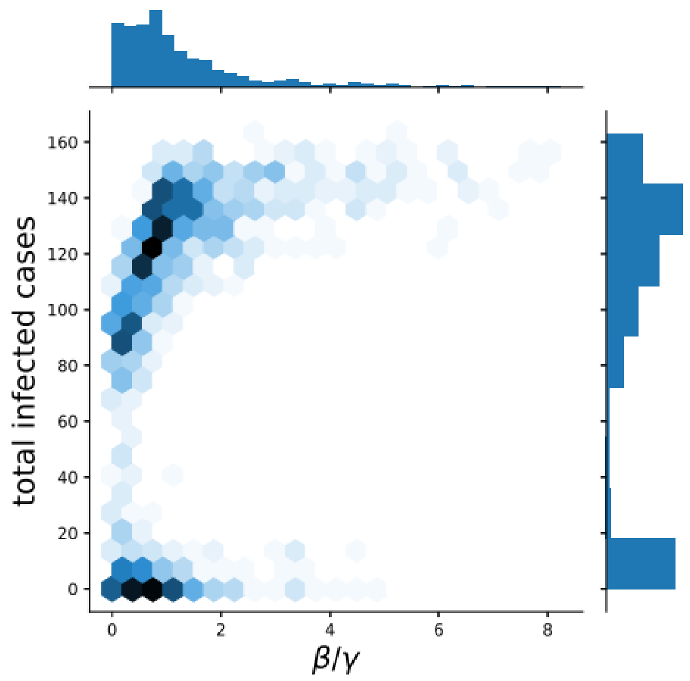
Learning maximizes the mutual information between local (node/patch) embeddings $\vec{h}_i$ and global graph summaries s

Use a proxy discriminator D to obtain probability scores for local-global pairs (and out of current graph patches for negative examples)

Dealing with Multimodal Graph Distributions



Errica, Bacciu, Micheli, ICML 2021

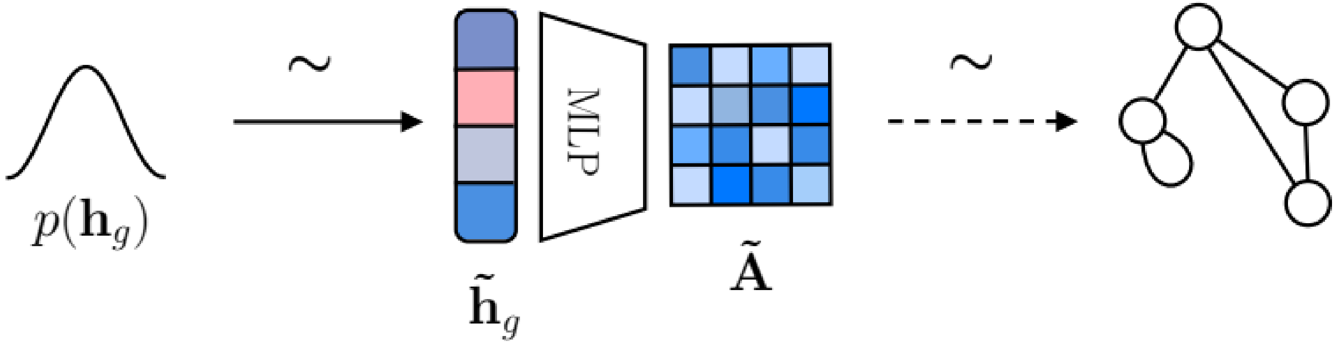


Graph Mixture Density Networks and their application to epidemiology

Generating Graphs

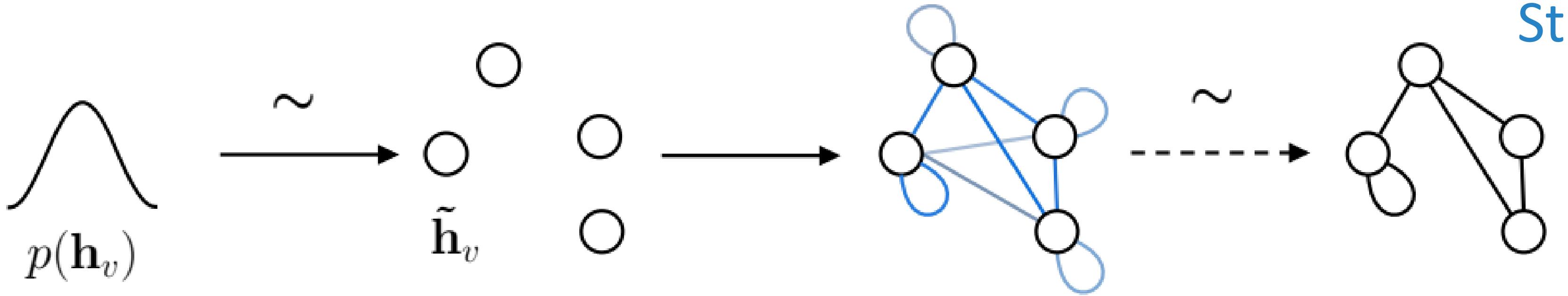
Graph Generation

Generate a prediction that is itself a graph

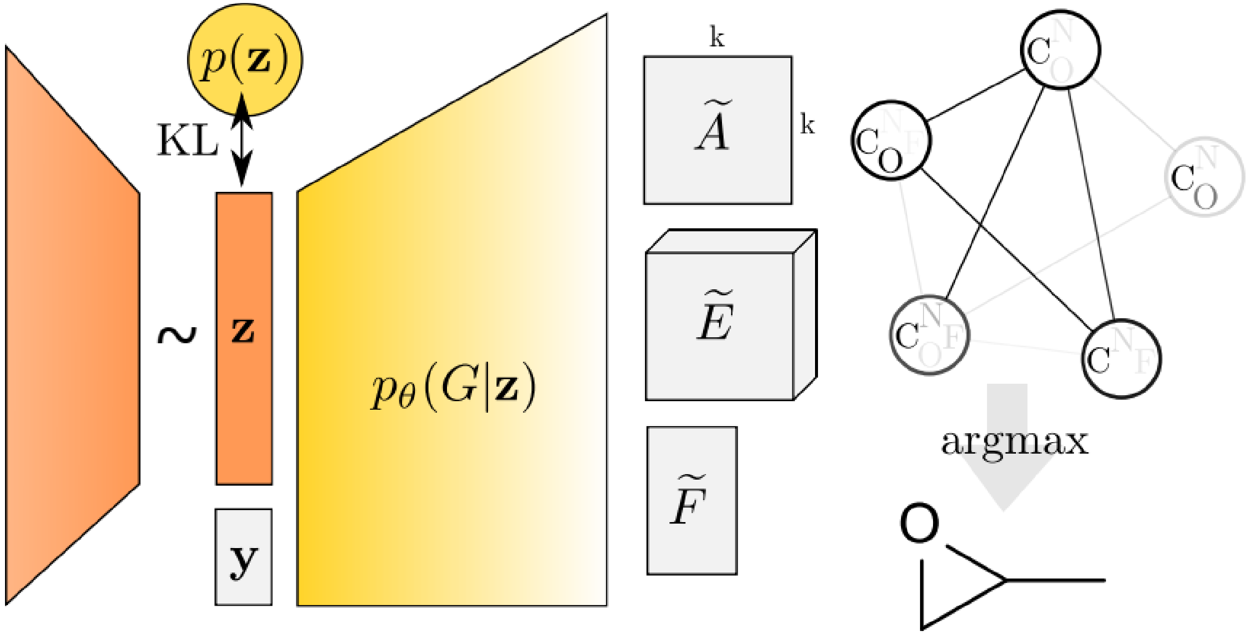


Adjacency-based

Structure-based



Graph Variational Autoencoder



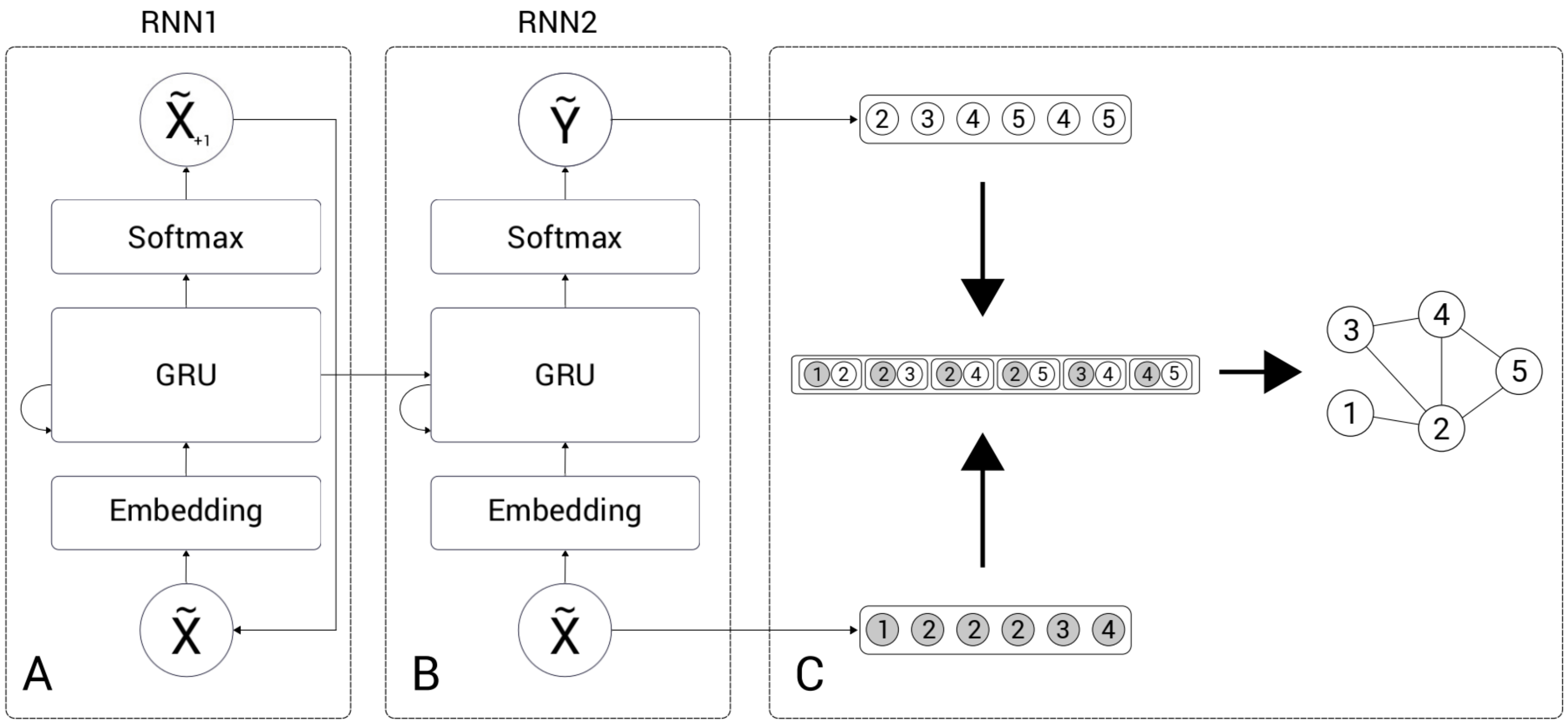
GraphVAE generates adjacency matrix up to k vertices

Sample molecules by latent space interpolation

Simonovsky, Komodakis, ICLR-WS 2018

Language-Based Graph (Structure) Generation

Generate a graph node-by-node and edge-by-edge through a sequential approach

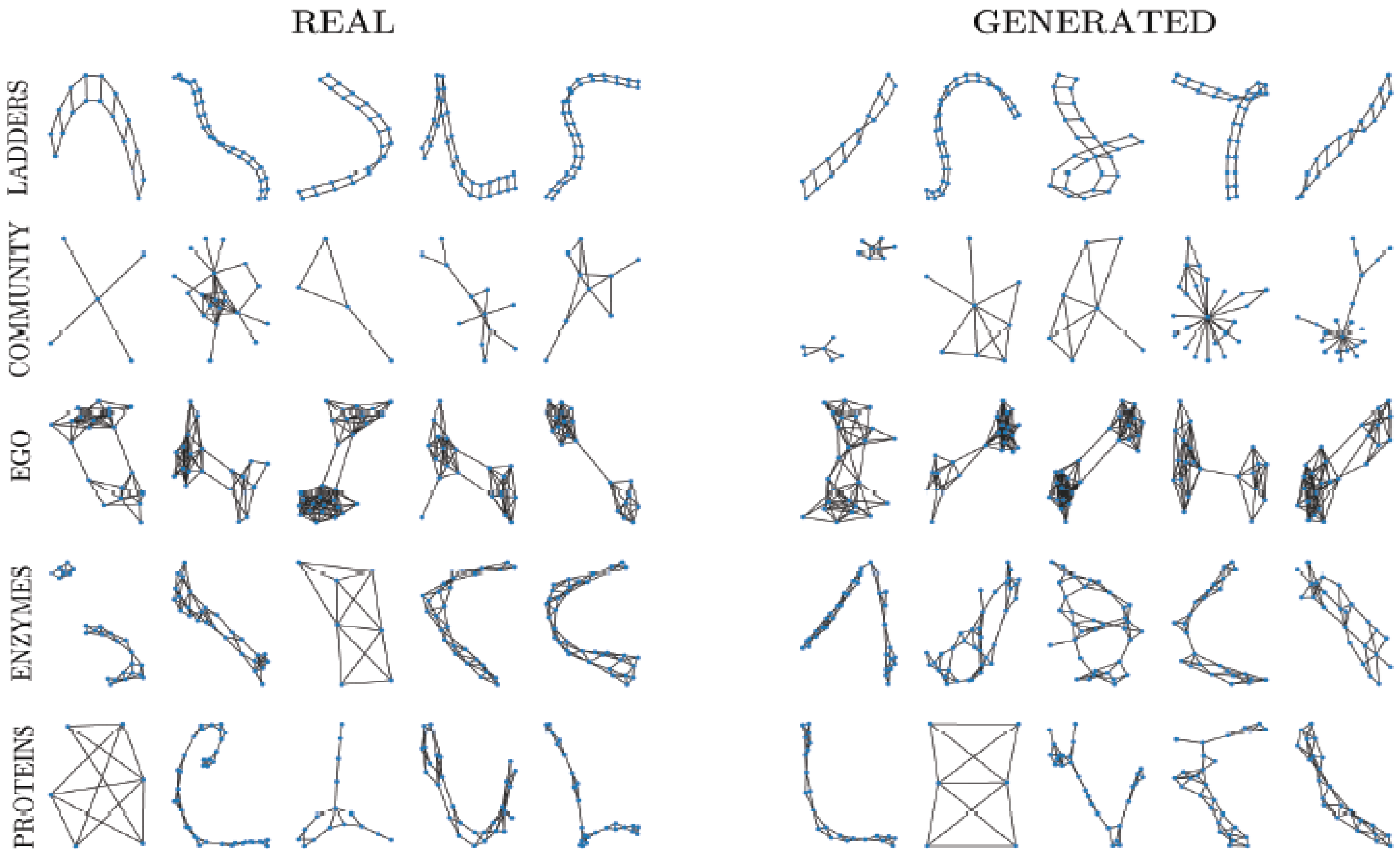


How to choose node ordering?

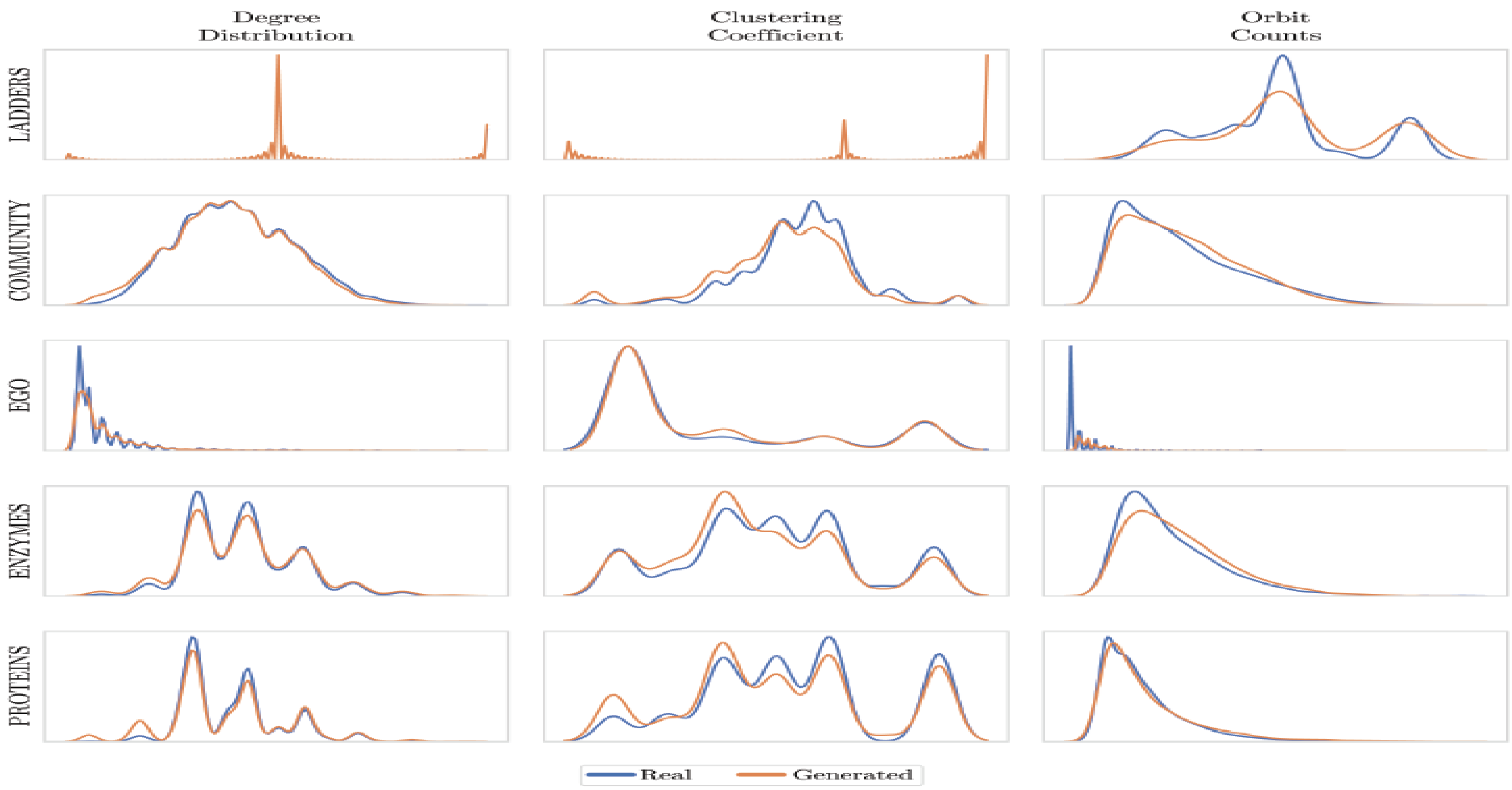
You et al, ICML 2018

Bacciu et al, Neurcomputing 2020

Lets generate some general structures…

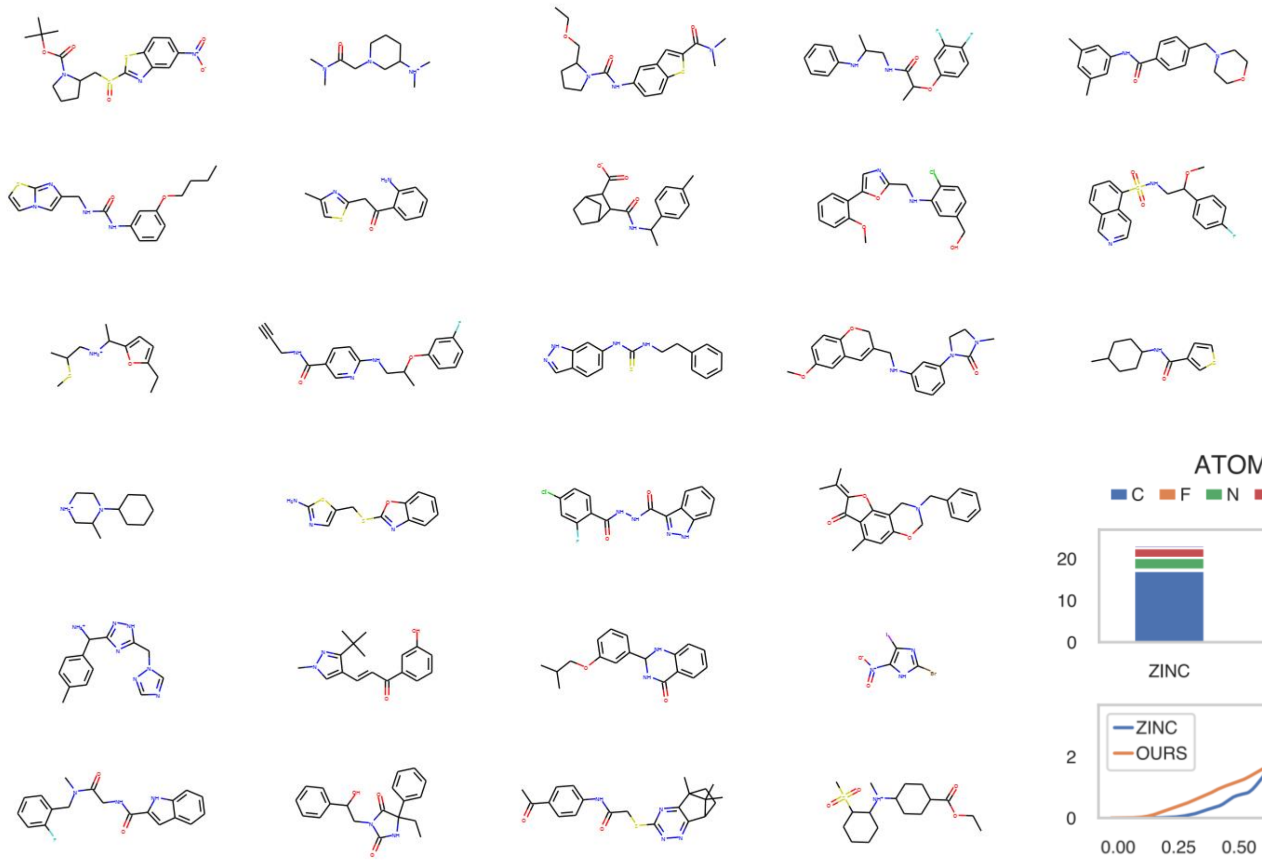


...with good structural properties



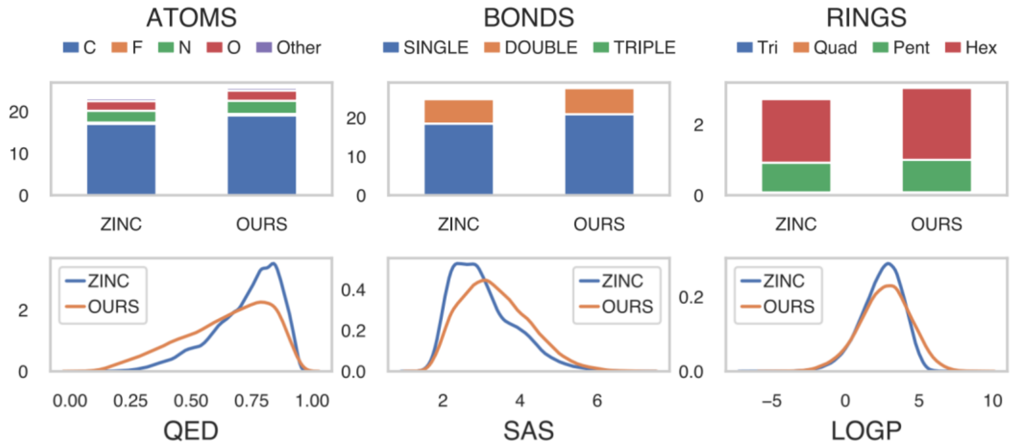
Bacciu et al,
Neurcomputing 2020

Generating Molecules



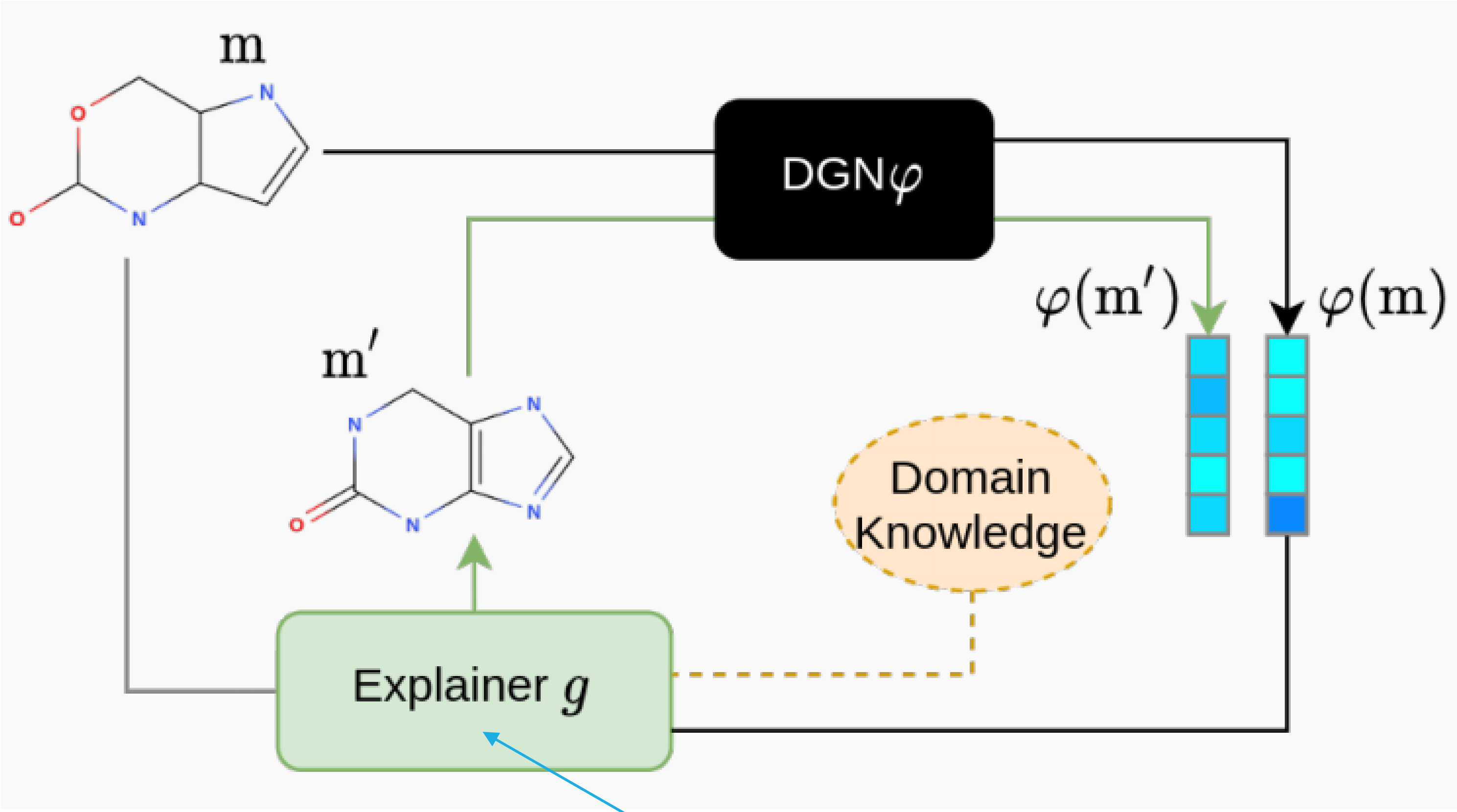
Fragment-based deep molecule generation

Podda, Bacciu, Micheli, AISTATS 2020



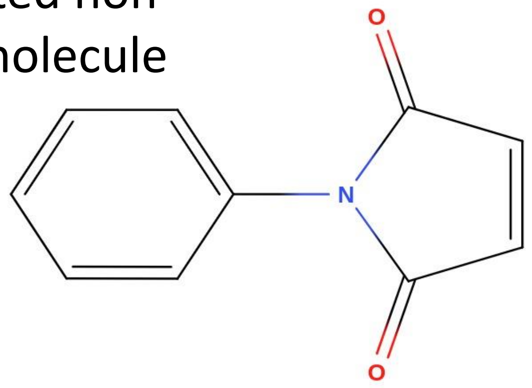
Generate Counterfactual Molecules for DGN Explainability

Bacciu et al, NeurIPS-WS 2020 / IJCNN 2021

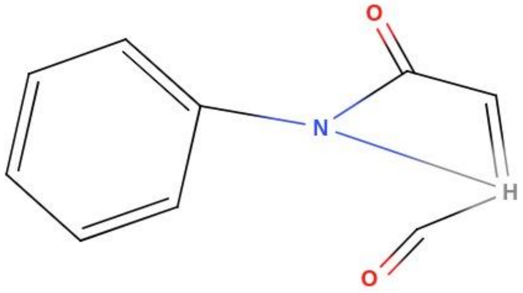


Reinforcement-learning based graph alterations

Predicted non-toxic molecule



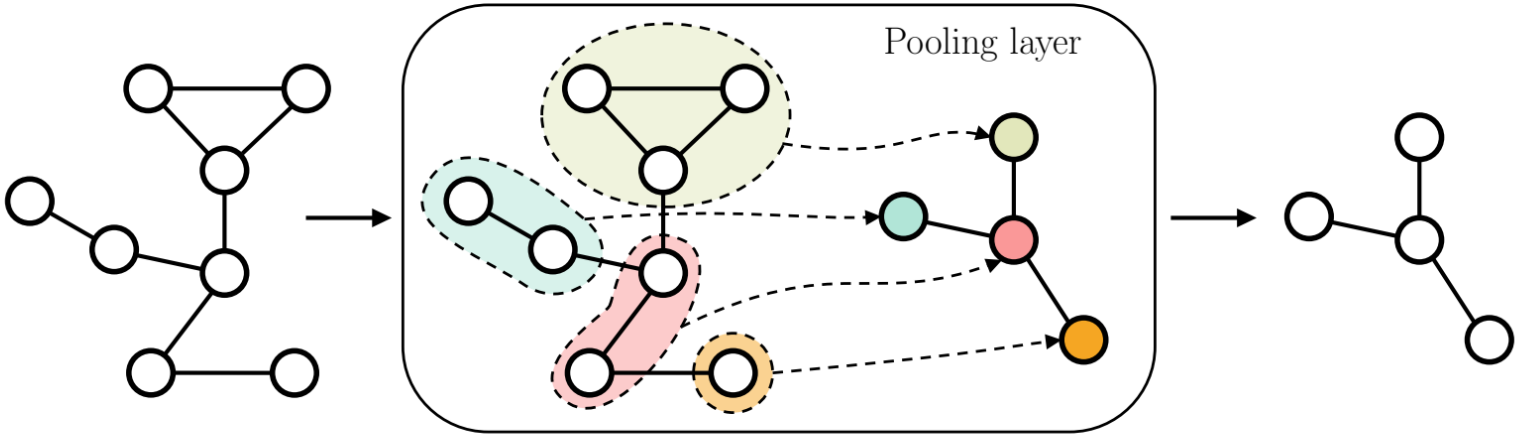
Counterfactual



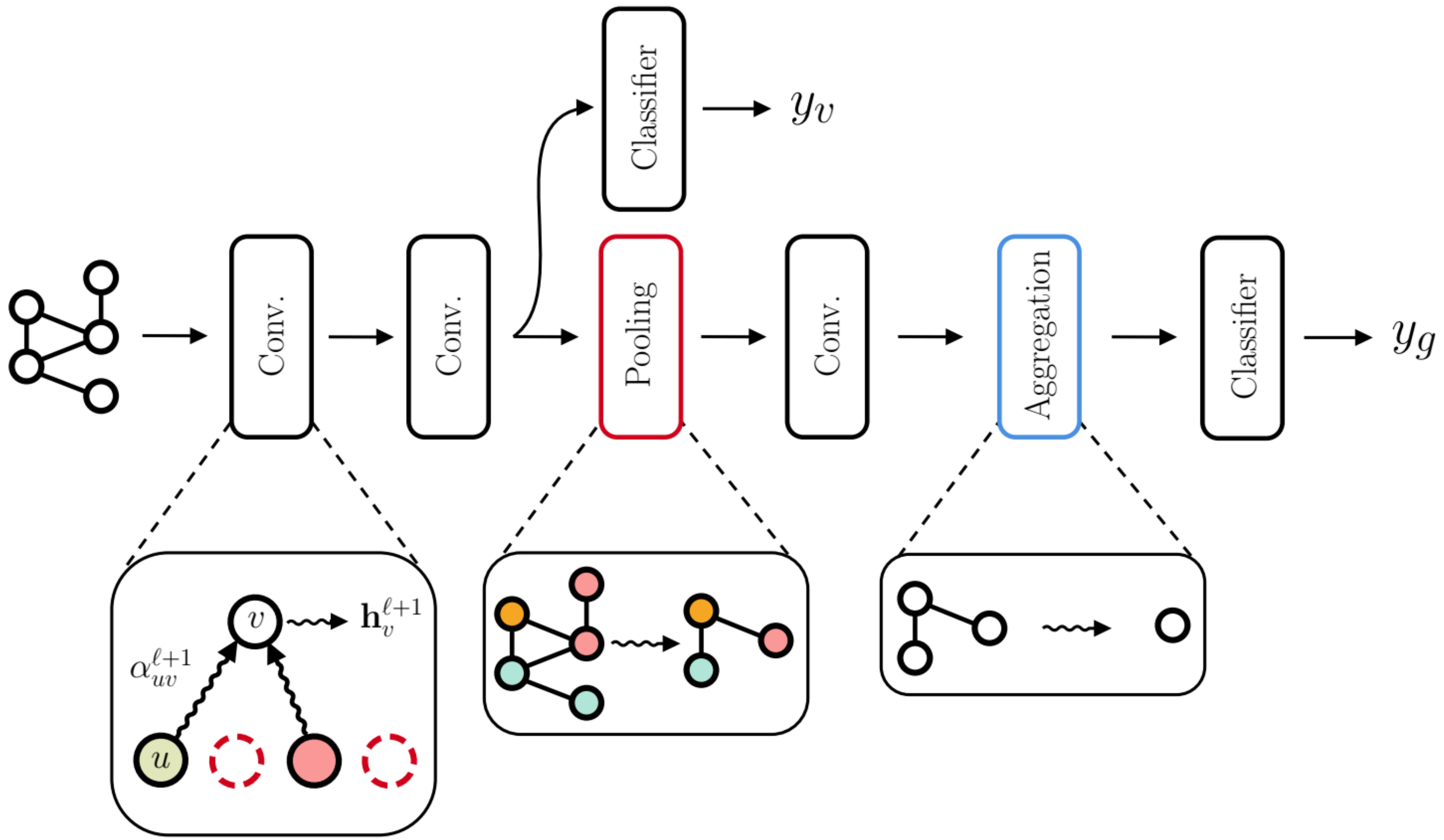
Advanced topics & research directions

What About Pooling?

- Standard aggregation operates of predefined node subsets
- Ignore community/hierarchical structure in the graph
- Need graph coarsening (pooling) operators
  - Differentiable
  - Graph theoretical
  - Graph signature

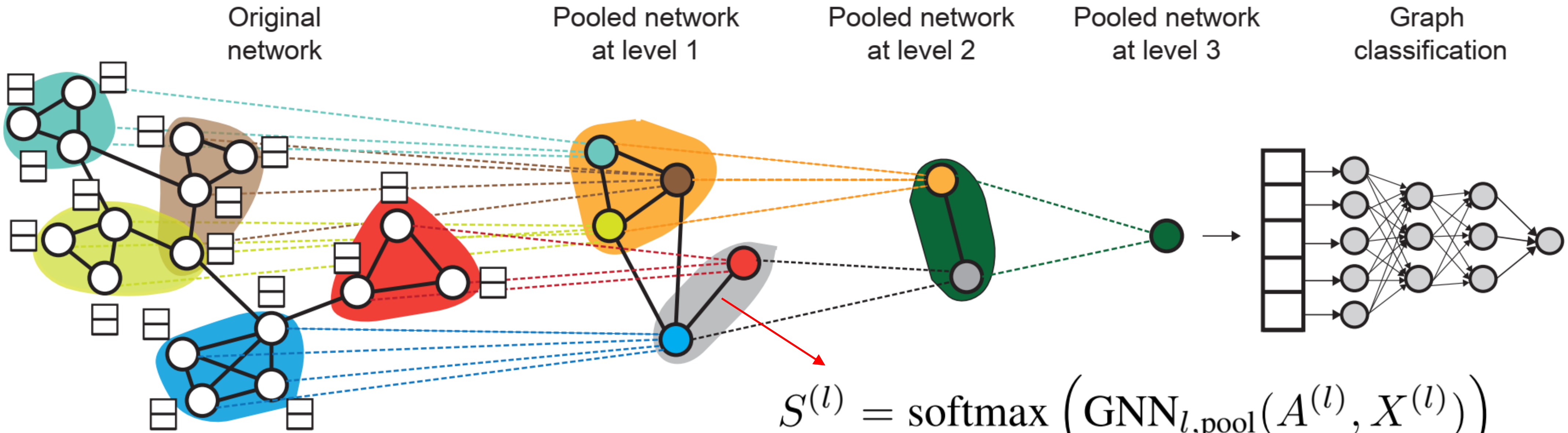


The Complete Picture – Graph Convolutions & Graph Pooling



Differentiable Graph Pooling - DiffPool

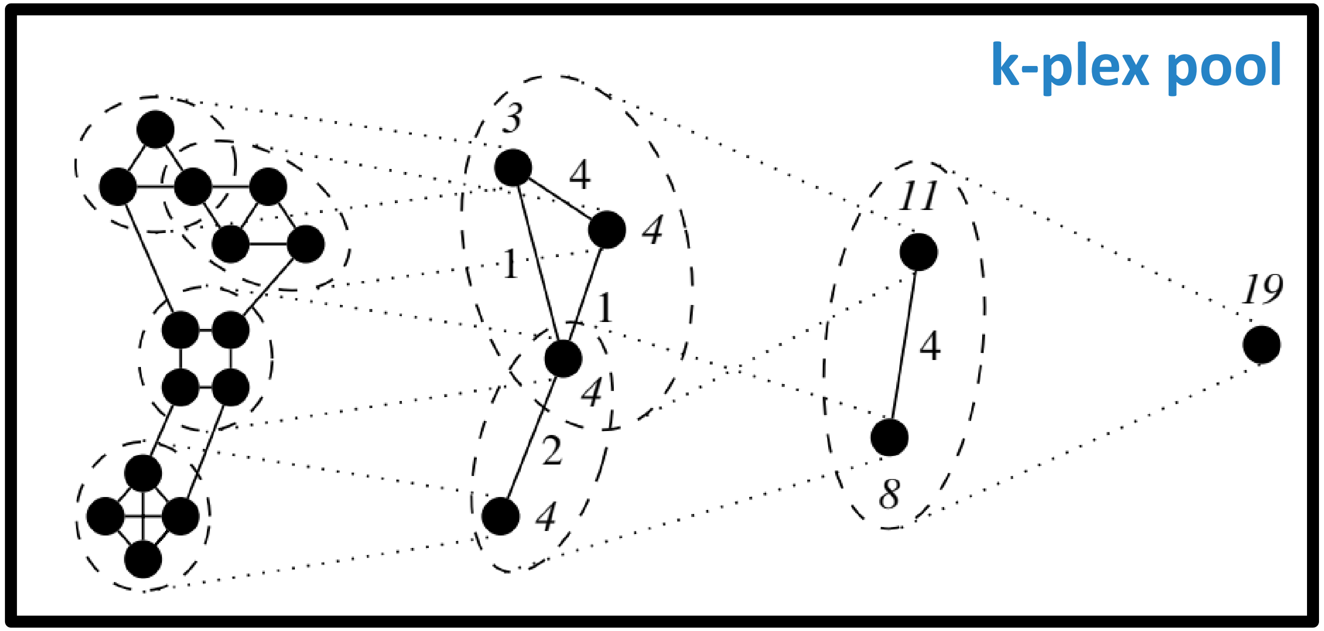
Rex Ying et al, NIPS 2018



$$S^{(l)} = \text{softmax}\left(\text{GNN}_{l,\text{pool}}(A^{(l)}, X^{(l)})\right)$$

GNN embedding followed by softmax to obtain a matrix of (probabilistic) assignment of nodes to clusters

Graph Theoretical Approaches



Identify local community structures in the graph

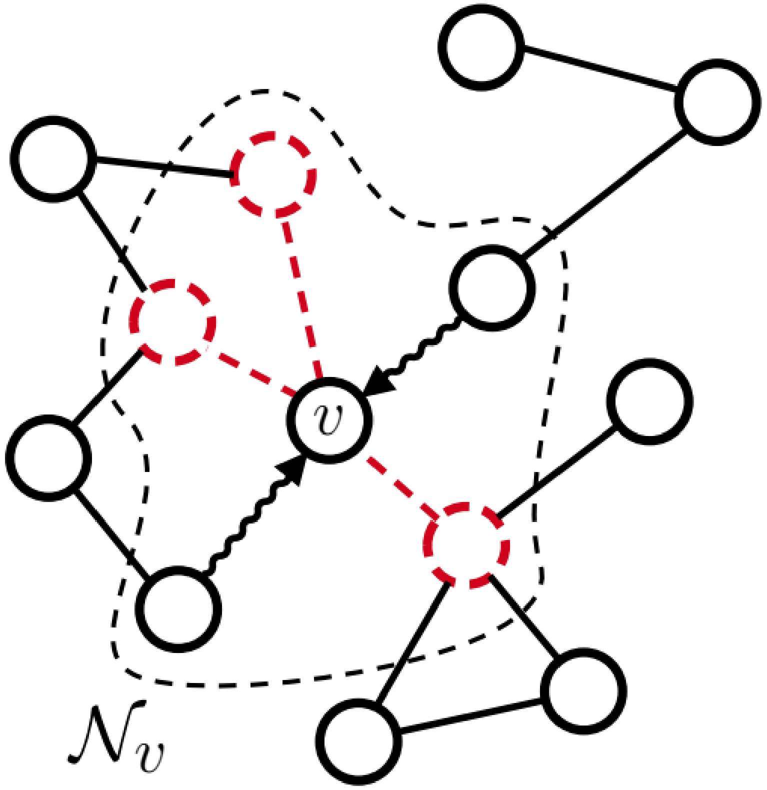
Algorithmic approaches

- Cliques (Velickovic et al, 2019)
- k-plex cover (Bacciu et al, NeurIPS-WS 2020, ECML/PKDD 2021)
- Max-ind set (Bacciu et al, 2021)

Factorization based

- Community discovery as non-negative matrix factorization (NMF)
- NMF-Pook - Bacciu & Di Sotto, 2019

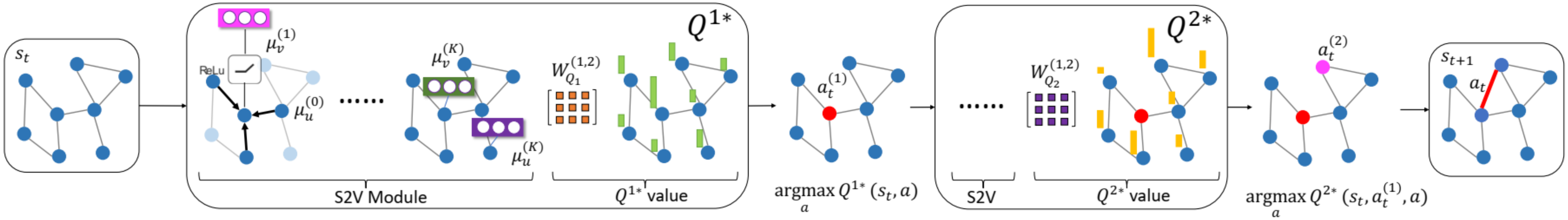
Scaling up to large graphs



- Dealing with large-scale graphs
  - Social networks
  - Recommendation systems
  - Biomedical network data
- How?
  - Sampling
  - Modularization (communities)
  - Active learning
  - HPC on graphs

Adversarial Attacks

Learn an attack policy by Q-learning (edge addition or deletion)

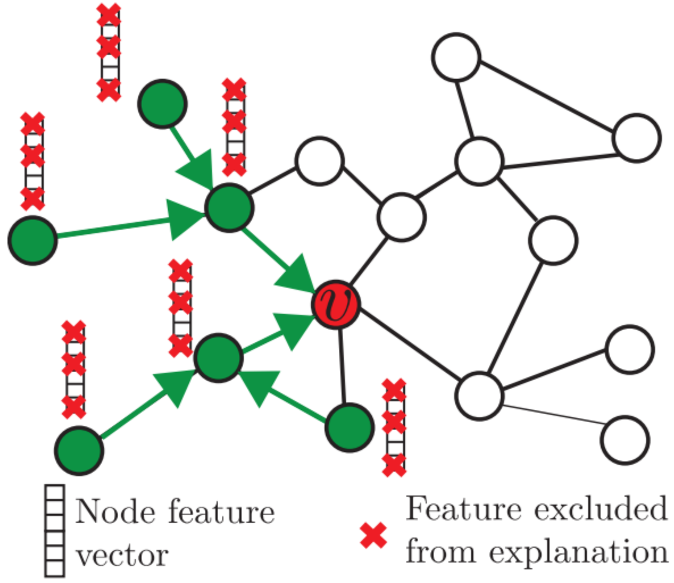


Show GraphRNN vulnerability to both black-box and white-box attacks (attack edges with maximum gradient)

H. Dai et al, ICML 2018

Interpretable Graph Networks

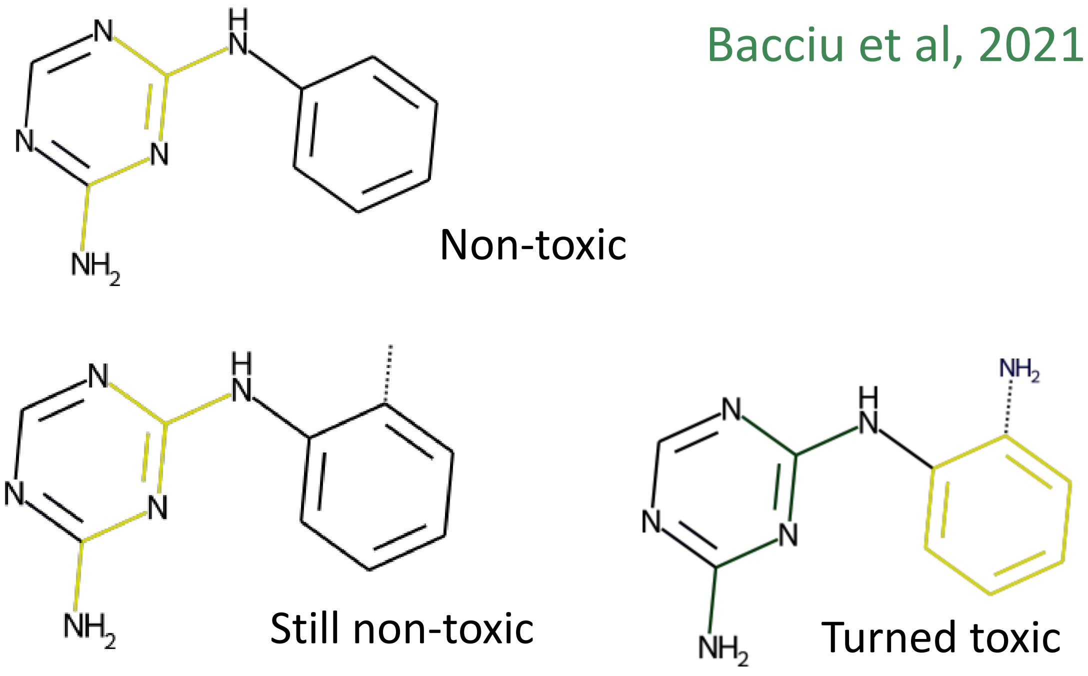
Identify relevant substructures and features for the prediction



R. Xing et al, NeurIPS 2019

Explain predictions locally with counterfactuals and local linear models

Bacciu et al, 2021



Reproducible Science (and Graphs)

Reproducibility Issues in Graph Classification

Lack of details/code
- Data preprocessing steps
- Features used
- Model selection
- Model evaluation/risk assessment

Experimental ambiguity
- Label stratification?
- Accuracy of model selection rather than risk assessment
  - Unclear if hyper-parameters optimized on the 10 **test** folds (**unfair**)
- Missing standard deviation

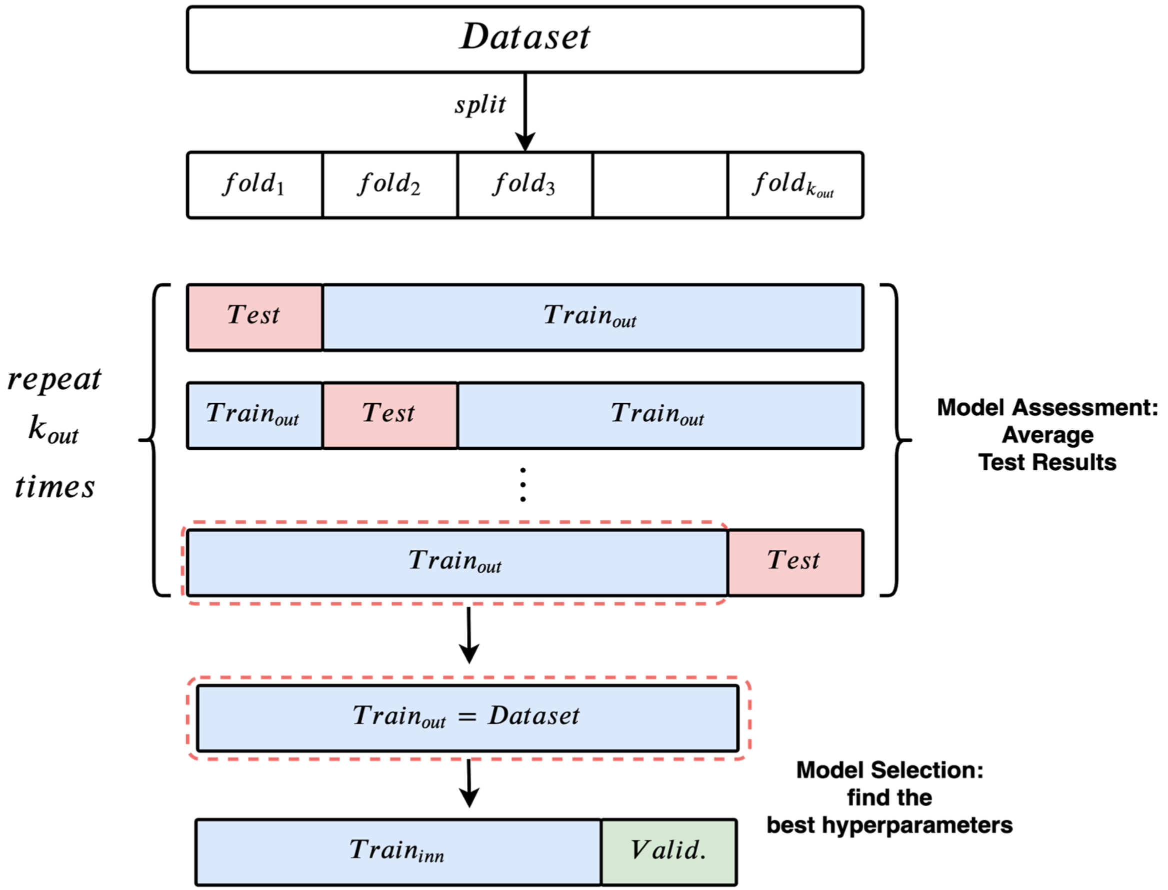
→ **Unclear, possibly unfair, and irreproducible experiments!**

A uniform empirical setting for assessing models in literature

10-fold CV for **risk assessment**
- Estimate of the true performances
- 10 test folds **never seen**

For each outer fold, simple hold-out for **model selection**
- **Validation** used to select "best" hyper-parameters

Run ≥ 47k experiments
- 5 models
- 9 datasets (chemical and social)



Results

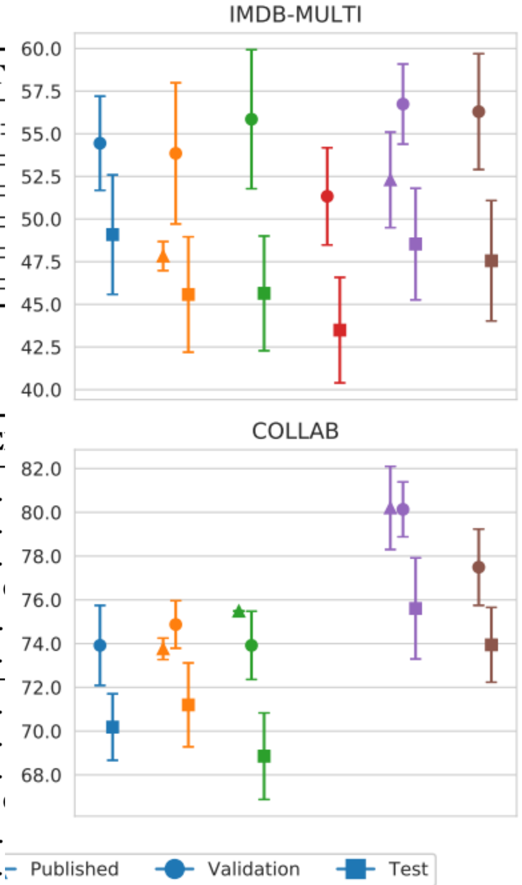
Errica et al, ICLR 2020

Chemical

	D&D	NCI1	PROT
Baseline	**78.4** ± 4.5	69.8 ± 2.2	**75.8**
DGCNN	76.6 ± 4.3	76.4 ± 1.7	72.9
DiffPool	75.0 ± 3.5	76.9 ± 1.9	73.7
ECC	72.6 ± 4.1	76.2 ± 1.4	72.3
GIN	75.3 ± 2.9	**80.0** ± 1.4	73.3
GraphSAGE	72.9 ± 2.0	76.0 ± 1.8	73.0

Social

		IMDB-B	IMDB-M	RE
NO FEATURES	Baseline	50.7 ± 2.4	36.1 ± 3.0	72.
	DGCNN	53.3 ± 5.0	38.6 ± 2.2	77.
	DiffPool	68.3 ± 6.1	45.1 ± 3.2	76.
	ECC	67.8 ± 4.8	44.8 ± 3.1	
	GIN	66.8 ± 3.9	42.2 ± 4.6	**87.**
	GraphSAGE	**69.9** ± 4.6	**47.2** ± 3.6	86.
WITH DEGREE	Baseline	70.8 ± 5.0	**49.1** ± 3.5	82.
	DGCNN	69.2 ± 3.0	45.6 ± 3.4	87.
	DiffPool	68.4 ± 3.3	45.6 ± 3.4	89.
	ECC	67.7 ± 2.8	43.5 ± 3.1	
	GIN	**71.2** ± 3.9	48.5 ± 3.3	**89.**
	GraphSAGE	68.8 ± 4.5	47.6 ± 3.5	84.



Main points:

- Baselines (no structure) can perform better!
  - **Chemical baseline**: Molecular fingerprint (Ralaivola et al., 2005)
  - **Social baseline**: node MLP + nodes aggregation + graph MLP
- Structure still needs to be **fully** exploited
- Node degree affects results

Software

You can find most of the foundational models in this tutorial implemented here

PyTorch geometric

DeepGraphLibrary

Python library for speeding up prototyping and reproducible Deep Graph Networks benchmarking

github.com/diningphil/PyDGN

PYDGN

Data (Benchmarks)

OPEN GRAPH BENCHMARK

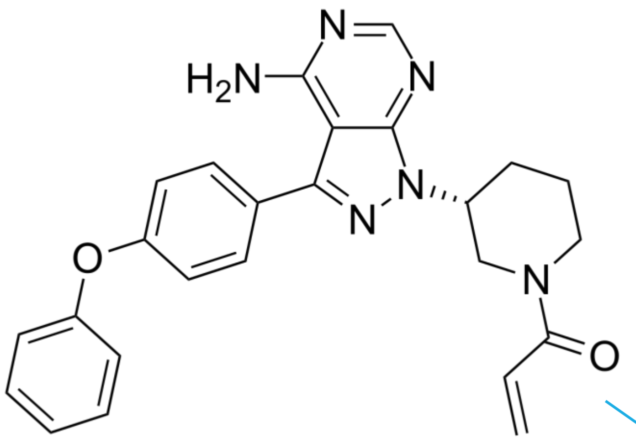
- Pytorch Geometric and DGL integration
- Standardized splits and evaluators + leader-board
- Node, link and graph property prediction tasks

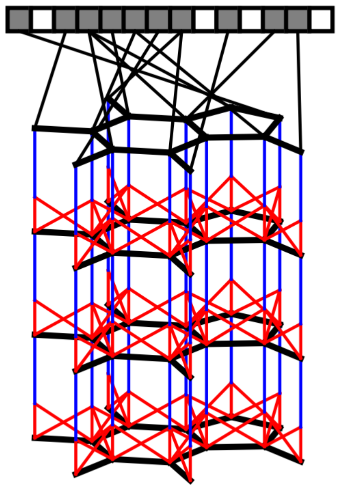
TUDataset

- Standardise assessment of existing benchmarks rather than inventing new ones
- Chemical, social, vision, synthetic, bioinformatics (with leader-board)
- Pytorch Geometric and DGL integration

Applications

Predicting Properties of Chemical Compounds





$\sim 10^{-2}$ seconds

Toxicity

Solubility

Quantum mechanical properties

Simulation methods

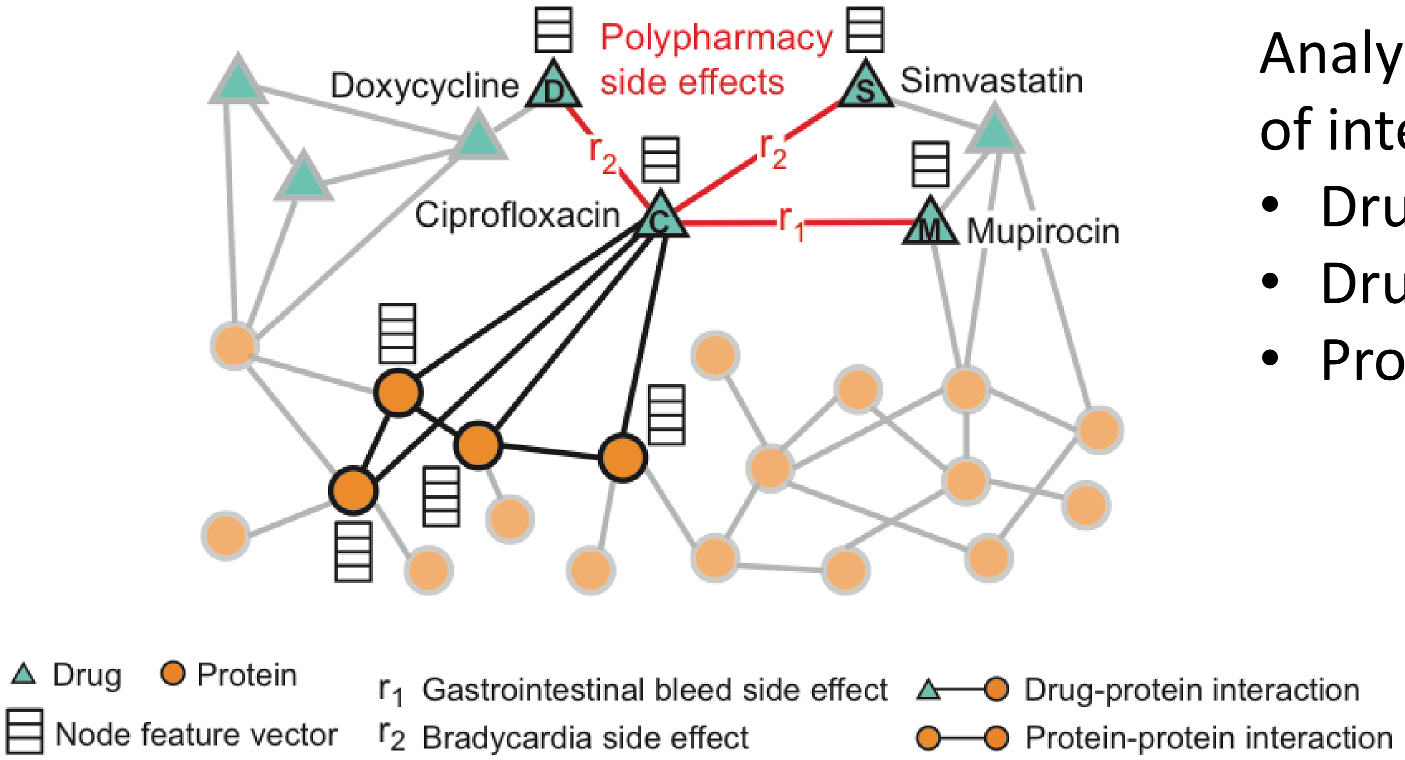
$\sim 10^{3}$ seconds

Micheli et al, JCICS 2001

Duvenaud, Maclaurin et al, NIPS 2015

Gilmer et al, ICML 2017

Side Effects of Drug Combinations

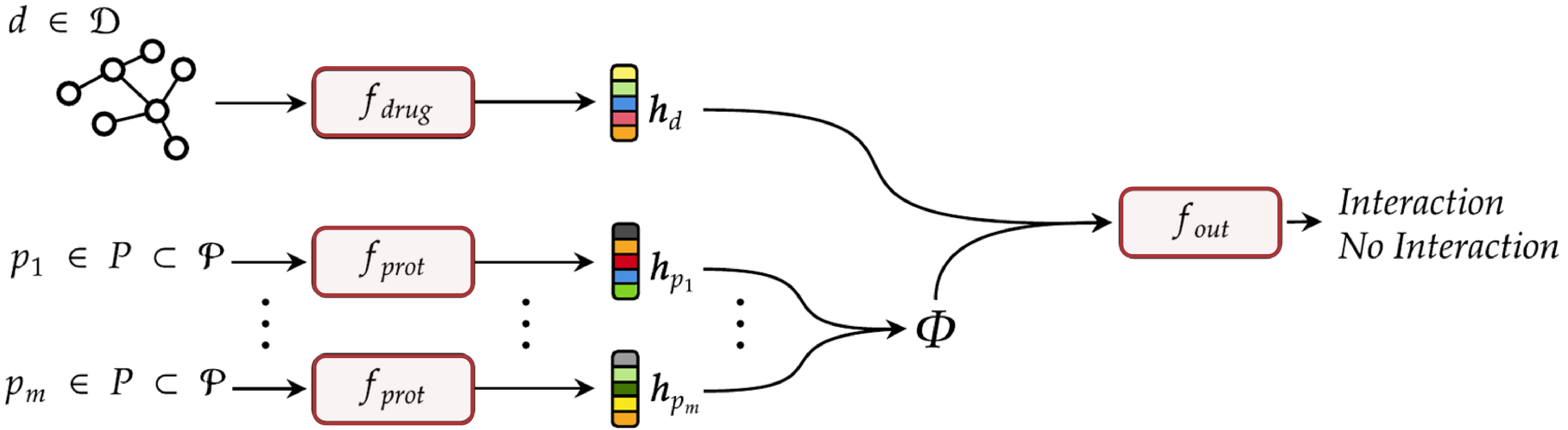


Analyzing a multimodal graph of interactions

- Drug-drug
- Drug-protein
- Protein-protein

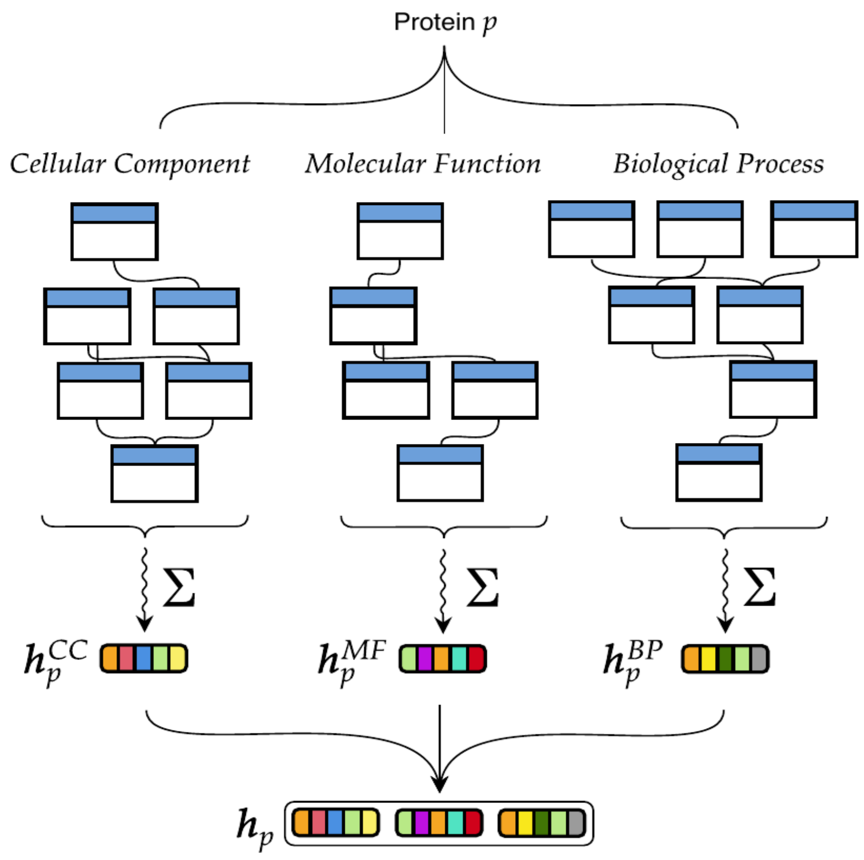
Zitnik, Agrawal, Leskovec, Bioinformatics 2018

Drug Repurposing with Deep Graph Networks

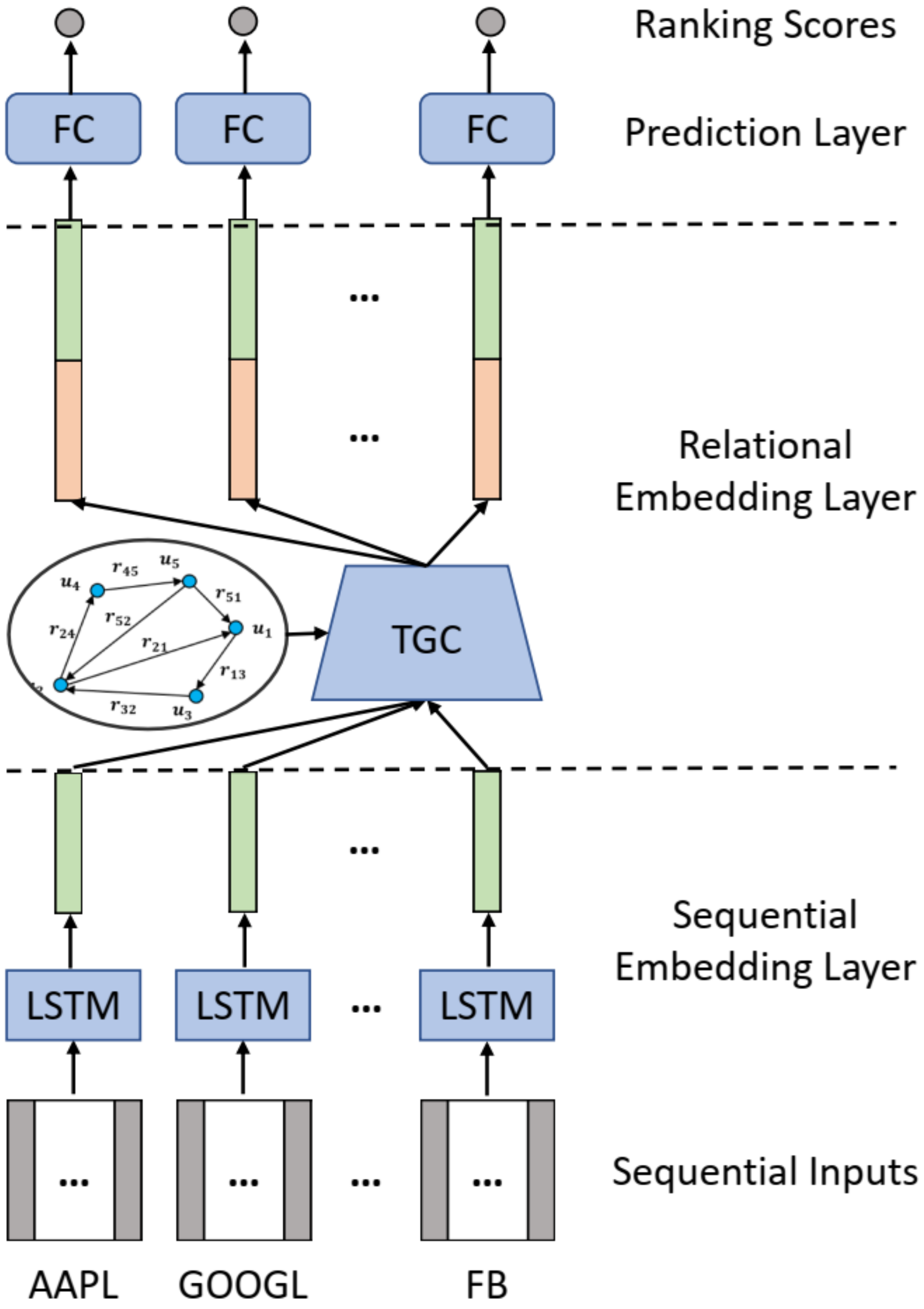


Protein embedding module

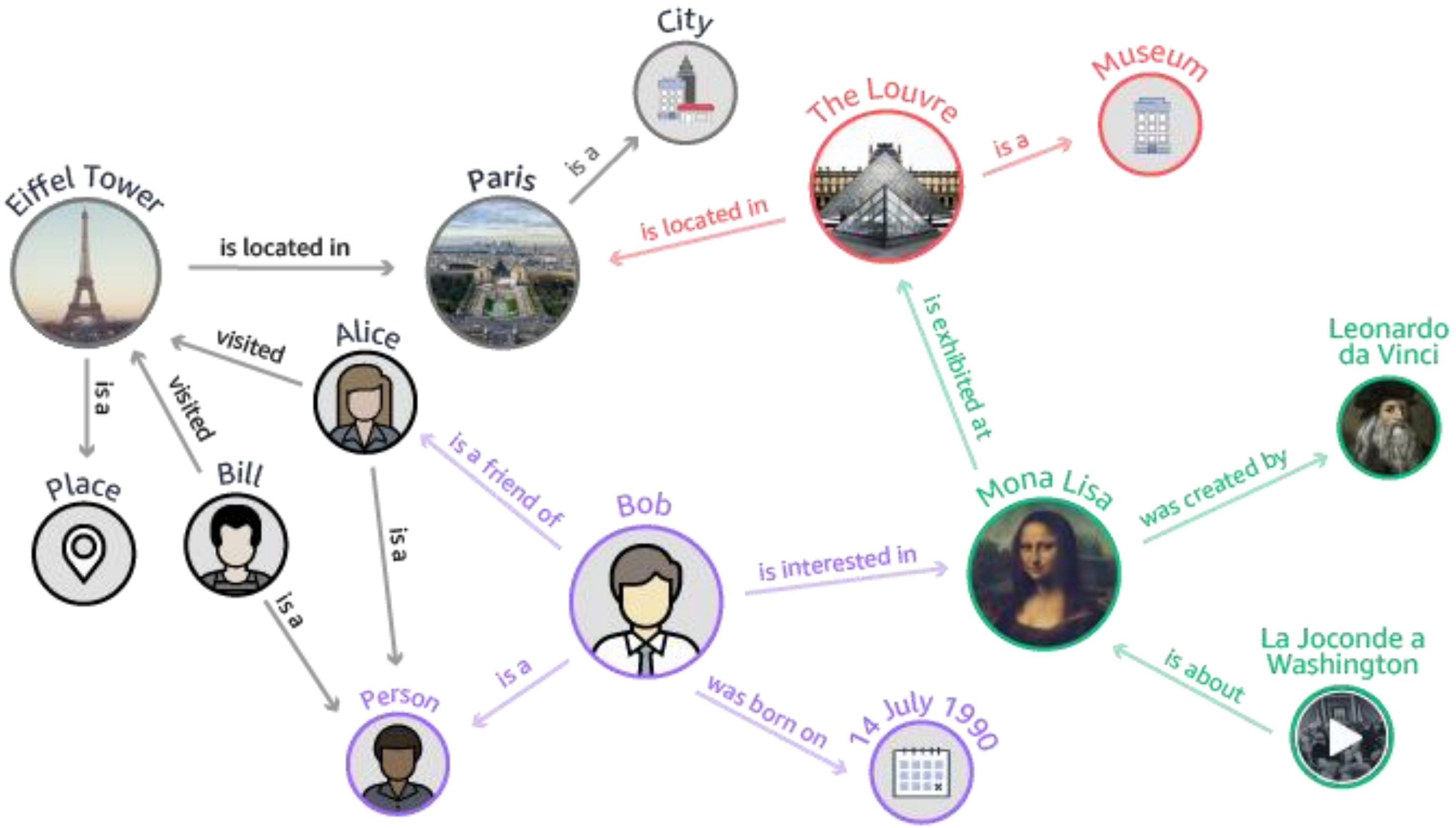
- Proteins represented by GO terms
- Applied Node2Vec to the 3 DAGs representing the GOs
- Pretrained module



Relational Stock Learning



Knowledge graphs

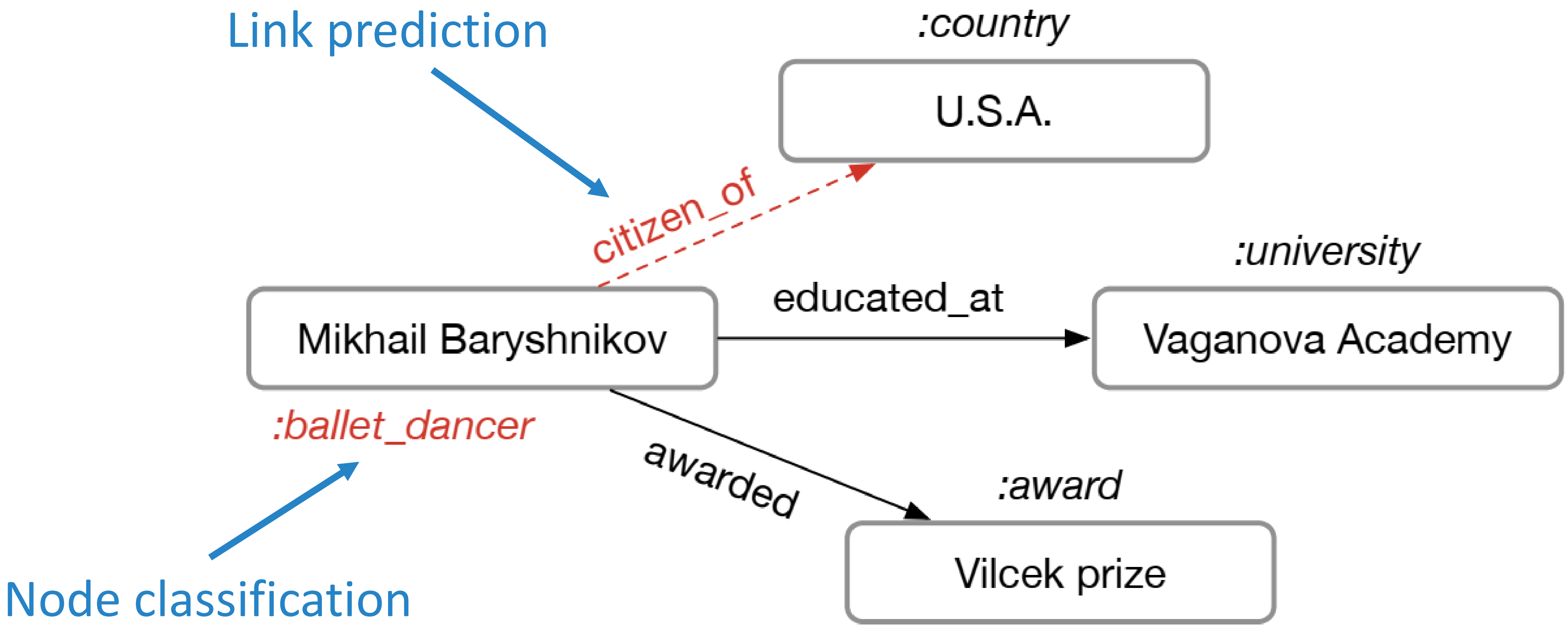


A natural way of representing known entities and relationships in a domain

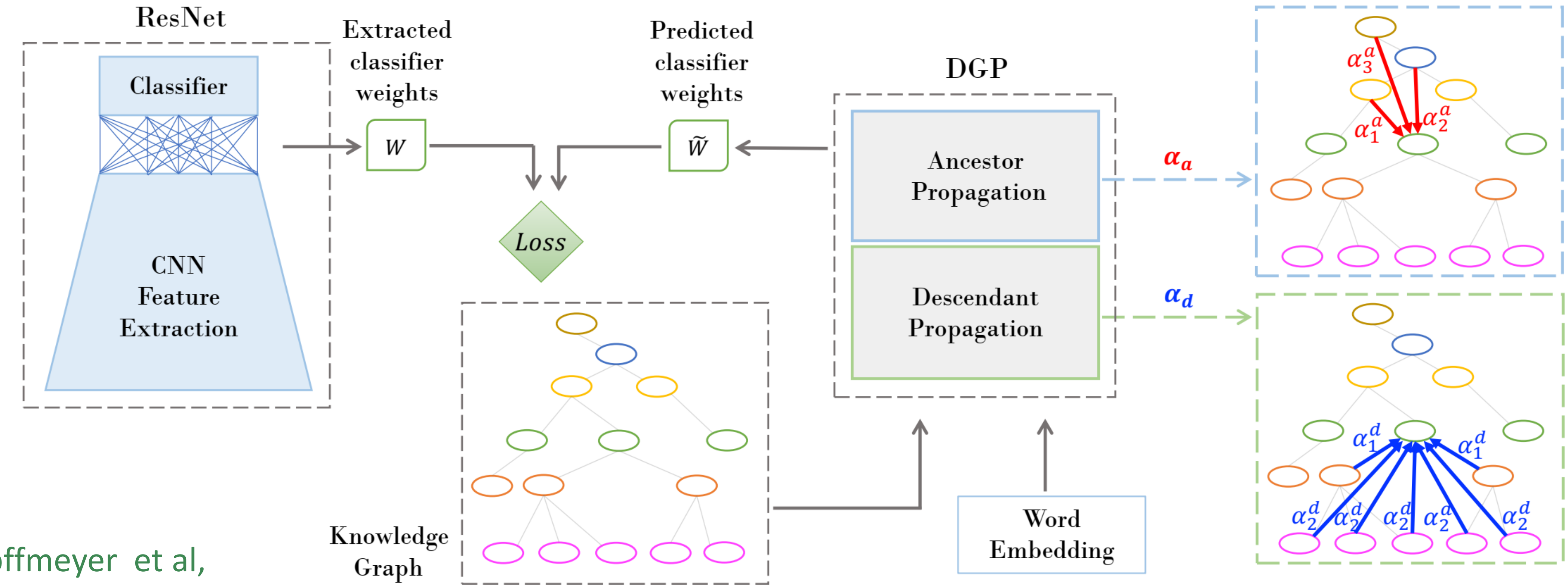
Node/link embeddings are numerical encodings of entities and relationships

Knowledge-based completion

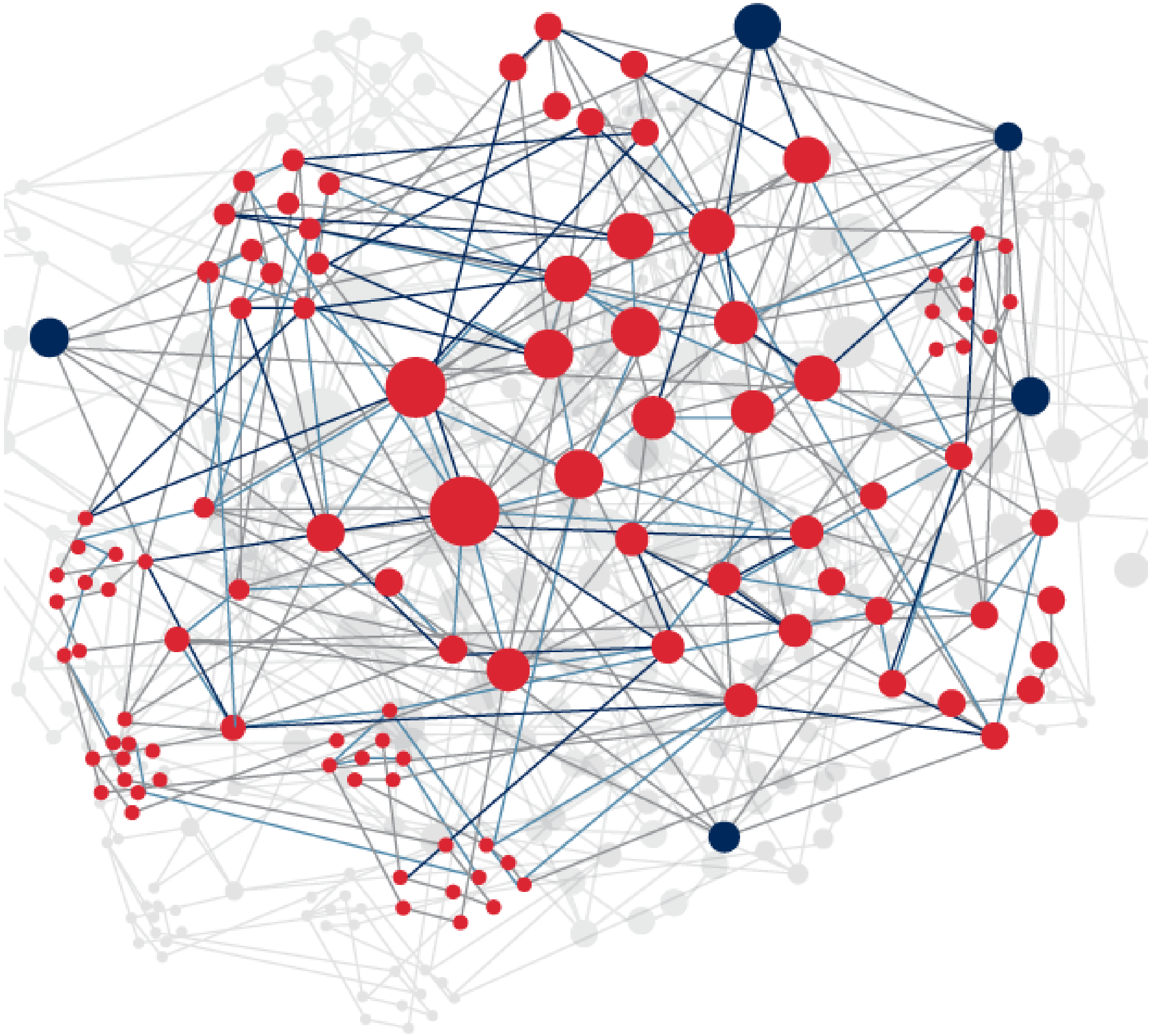
Schlichtkrull et al, ESWC 2018



Few Shot Learning with Knowledge Graphs



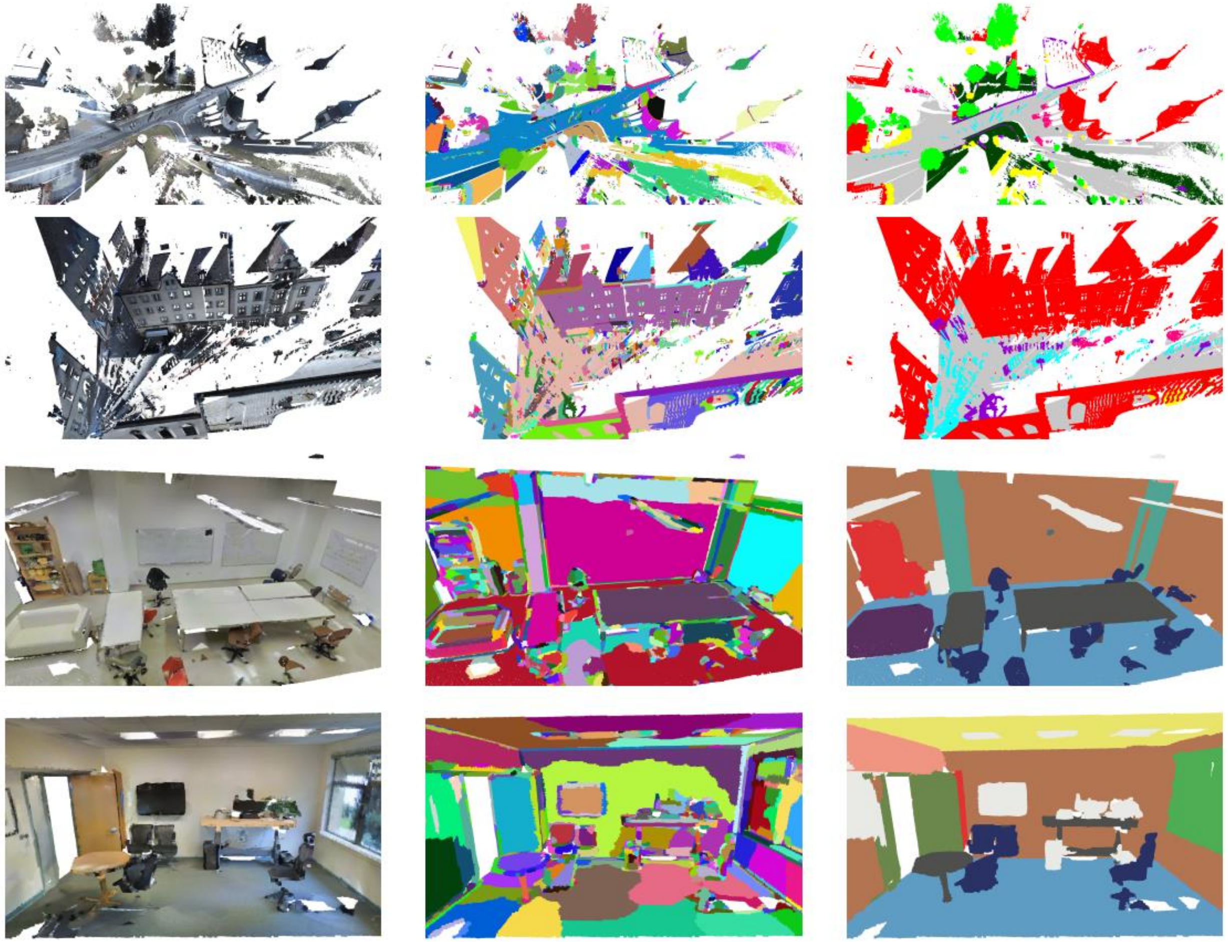
Kampffmeyer et al, CVPR 2018



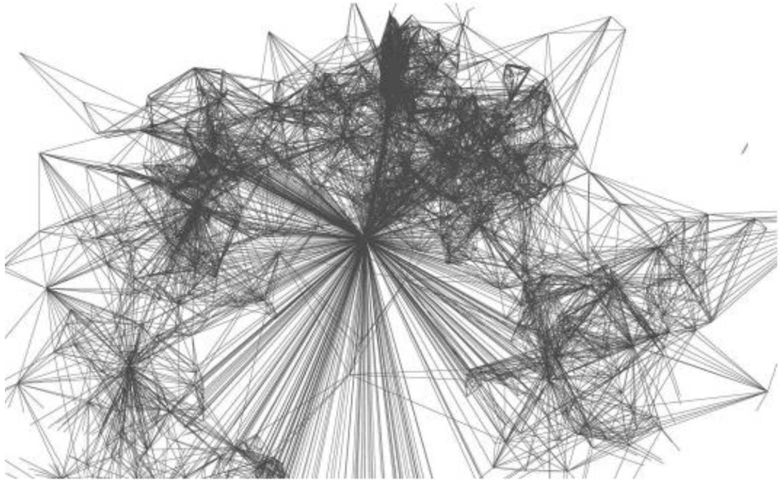
Recommendation Systems

…and other kinds of social network analyses

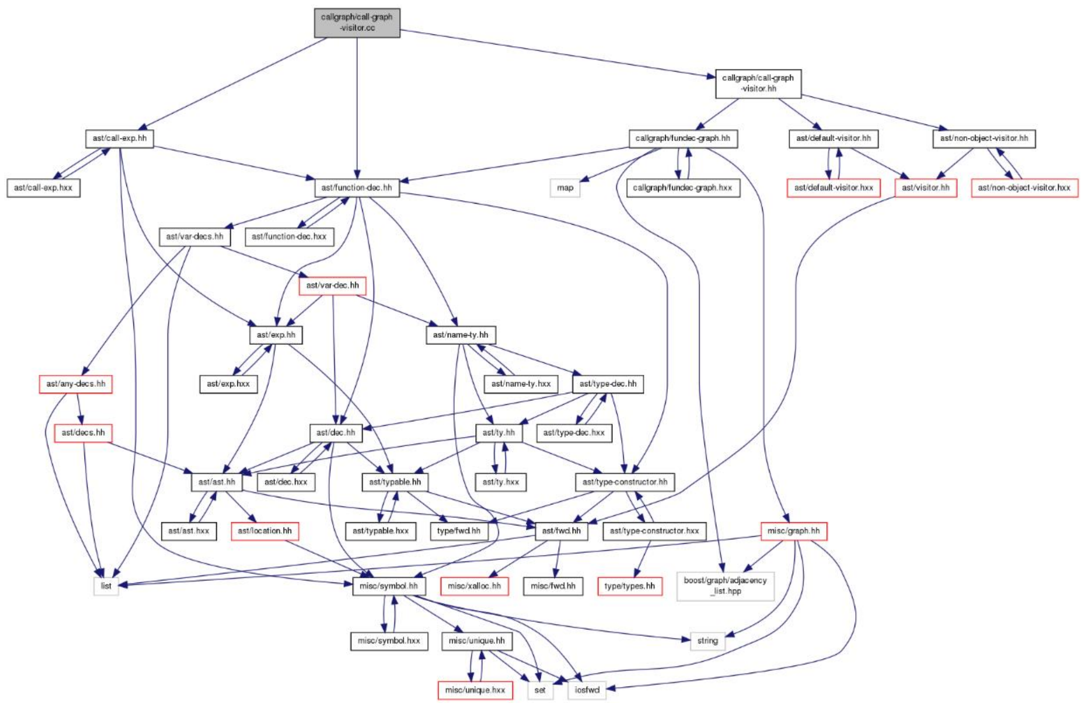
Point Clouds – Semantic Segmentation

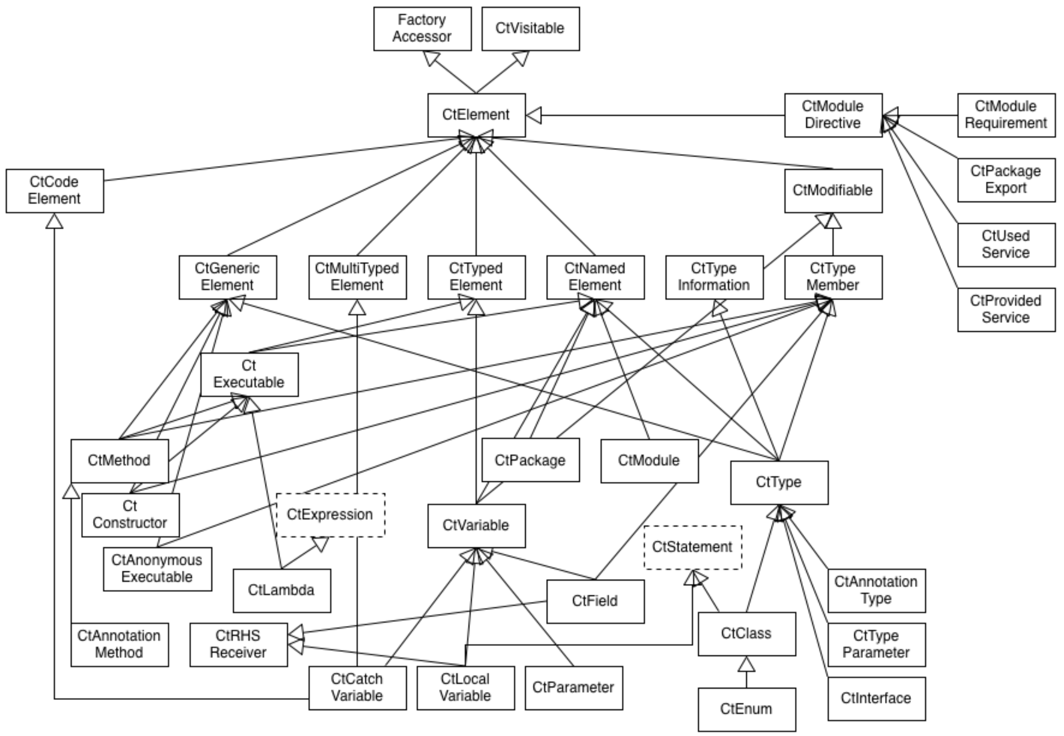


Build point cloud graphs and train semantic class predictors based on vertex embeddings



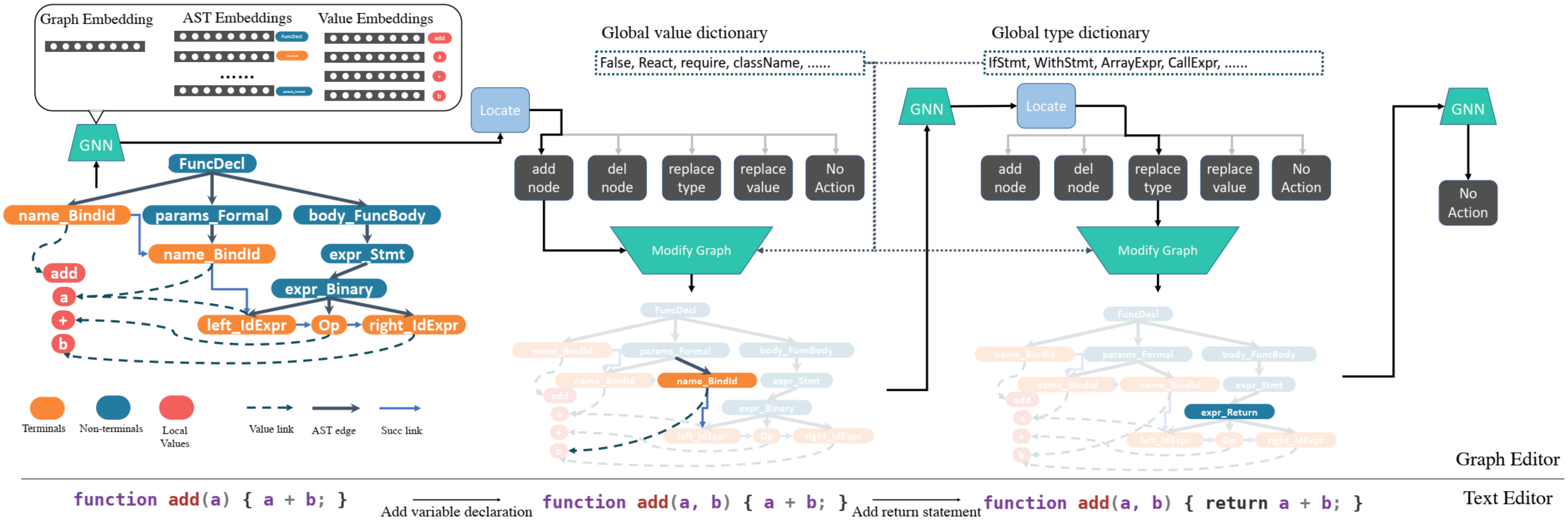
Landrieu, Simonovsky, CVPR 2018





Analysis of ICT systems

Code Correction as Graph Operations



Conclusions

- Deep learning for graphs is a research topic that is entering its consolidation phase
  - Many works sharing same underlying idea (adjacency, contractive, contextual)
  - Much early work left unacknowledged and reinvented
- What should we focus on?
  - Theoretical characterization and properties of operators (machine learning + graph theory)
  - Efficiency and efficacy of context creation and propagation (unsupervised, gradient issues, reinforcement learning & graphs)
  - Research directions (pooling, generative, transduction, expressivity, scalability, interpretability)
  - Applications (biomedical, software and ICT systems, large scale interaction networks)
- A candidate AI model for the integration of symbolic knowledge and numerical data

News and Ack’s

ESANN 2021 Special Session on Deep Learning for Graphs

October 8-10 2021

Co-organized by: C. Alippi, D. Bacciu, F.M Bianchi, B. Paassen

IEEE NNTC Task Force on Learning for Structured Data

Chair: D. Bacciu (bacciu@di.unipi.it) – ViceChairs: Filippo Maria Bianchi, Lorenzo Livi

www.learning4graphs.org

Promote events, research and dissemination activities for the community working on machine learning for structured data.

Advertisement Time

A tutorial paper reviewing the deep learning for graph area

D. Bacciu, F. Errica, A. Micheli, M. Podda, A Gentle Introduction to Deep Learning for Graphs, Neural Networks, 2020, Arxiv

PyDGN

Our Python library for Deep Graph Networks

github.com/diningphil/PyDGN

Upcoming Tutorials

IJCNN 2021 (23 July) – Deep learning for Graphs

Thank you!

DAVIDE BACCIU (BACCIU@DI.UNIPI.IT)

DIPARTIMENTO DI INFORMATICA - UNIVERSITA' DI PISA

Bibliography (I)

Spectral Domain Convolutions

1. Joan Bruna, Wojciech Zaremba, Arthur Szlam, Yann LeCun , Spectral Networks and Locally Connected Networks on Graphs, ICLR 2014
2. Mikael Henaff, Joan Bruna, Yann LeCun, Deep Convolutional Networks on Graph-Structured Data, Arxiv 2015
3. Michaël Defferrard, Xavier Bresson, Pierre Vandergheynst, Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering, NIPS 2016
4. Thomas N. Kipf, Max Welling, Semi-Supervised Classification with Graph Convolutional Networks, ICLR 2017

Spatial Domain Convolutions

1. David Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, Ryan P. Adams, Convolutional Networks on Graphs for Learning Molecular Fingerprints, NIPS 2015
2. Mathias Niepert, Mohamed Ahmed, Konstantin Kutzkov, Learning Convolutional Neural Networks for Graphs , ICML 2016

Bibliography (II)

Contextual Approaches

1. Alessio Micheli, Neural Network for Graphs: A Contextual Constructive Approach. IEEE TNN, 2009
2. Yujia Li, Daniel Tarlow, Marc Brockschmidt, Richard Zemel , Gated Graph Sequence Neural Networks, ICLR 2016
3. William L Hamilton, Rex Ying, Jure Leskovec, Inductive Representation Learning on Large Graphs, NIPS 2017.
4. Xu et al.: How Powerful are Graph Neural Networks?, ICLR 2019
5. Davide Bacciu, Federico Errica, Alessio Micheli , Contextual Graph Markov Model: A Deep and Generative Approach to Graph Processing, ICML 2018
6. Veličković et al, Deep Graph Infomax , ICLR 2019
7. Davide Bacciu, Federico Errica, Alessio Micheli , Probabilistic Learning on Graphs via Contextual Architectures, JMLR 2020

Bibliography (III)

Miscellanea

1. Davide Bacciu, Federico Errica, Alessio Micheli, Marco Podda, A Gentle Introduction to Deep Learning for Graphs, Neural Networks, 2020 (Tutorial)
2. Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, George E. Dahl, Neural Message Passing for Quantum Chemistry, ICML 2017 (Framework)
3. Martin Simonovsky, Nikos Komodakis, GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders, NIPS Workshop, 2017 (Graph Generation)
4. Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, Jure Leskovec, GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models, ICML 2018 (Graph Generation)
5. Davide Bacciu, Alessio Micheli, Marco Podda, Edge-based sequential graph generation with recurrent neural networks, Neurocomputing 2020 (Graph Generation)
6. Marco Podda, Davide Bacciu, Alessio Micheli, A Deep Generative Model for Fragment-Based Molecule Generation, AISTATS 2020 (Molecule Generation)

Bibliography (IV)

Miscellanea

1. Ying et al.: Hierarchical Graph Representation Learning with Differentiable Pooling, NeurIPS 2018 (Pooling)
2. Zhang et al, An End-to-End Deep Learning Architecture for Graph Classification, AAAI 2018 (Pooling)
3. D. Bacciu et al, K-plex Cover Pooling for Graph Neural Networks, NeurIPS WS 2020 (Pooling)
4. D. Bacciu, L. Di Sotto. A non-negative factorization approach to node pooling in graph convolutional neural networks Conference. AIIA 2019, (Pooling)
5. Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, Yoshua Bengio, Graph Attention Networks, ICLR 2018 (Attention)
6. Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, Le Song, Adversarial Attack on Graph Structured Data, ICML 2018 (Adversarial Attacks)
7. R. Ying et al, GNNExplainer: Generating Explanations for Graph Neural Networks, NeurIPS 2019 (Intepretability)
8. F Errica, D Bacciu, A Micheli, Graph Mixture Density Networks, ICML 2021 (Multimodal)