Deep Learning for Graphs Advanced

Generative and Deep Learning (GDL)

Davide Bacciu (davide.bacciu@unipi.it)

UNIVERSITÀ DI PISA

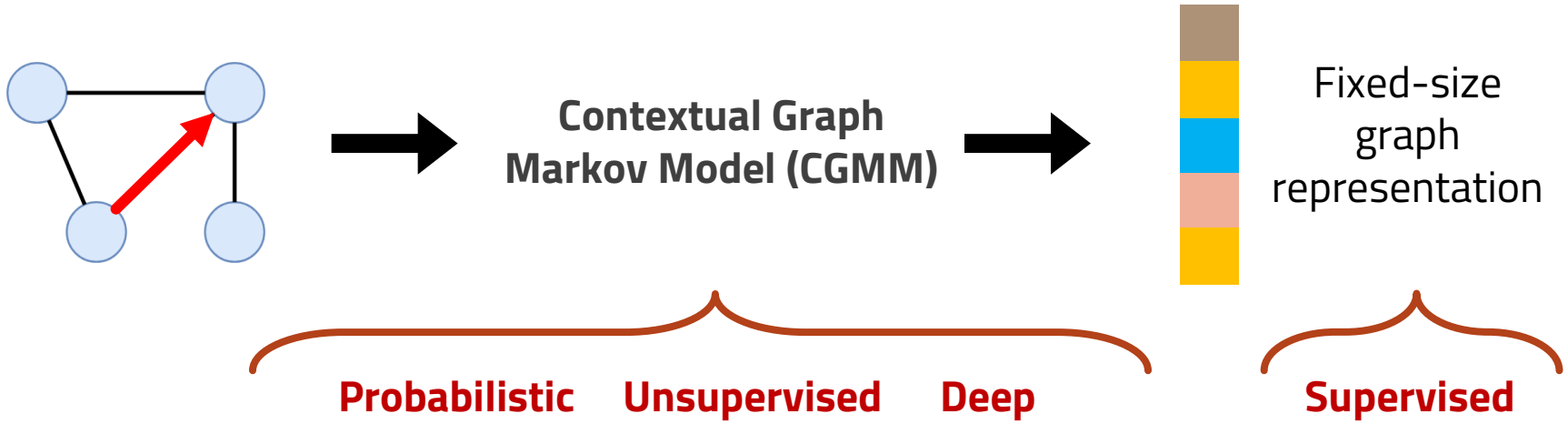
Objectives

- Generative graph learning
  - Probabilistic models on graphs
  - Graph VAE, graph language models and graph diffusion models
- Issues with information propagation on graphs
  - Oversmoothing, oversquashing and undereaching
  - Topological approaches
  - Dynamical systems approaches
- Spatio-temporal and dynamic graphs
- Neural Algorithmic Reasoning
- Applications

Probabilistic Graph Models

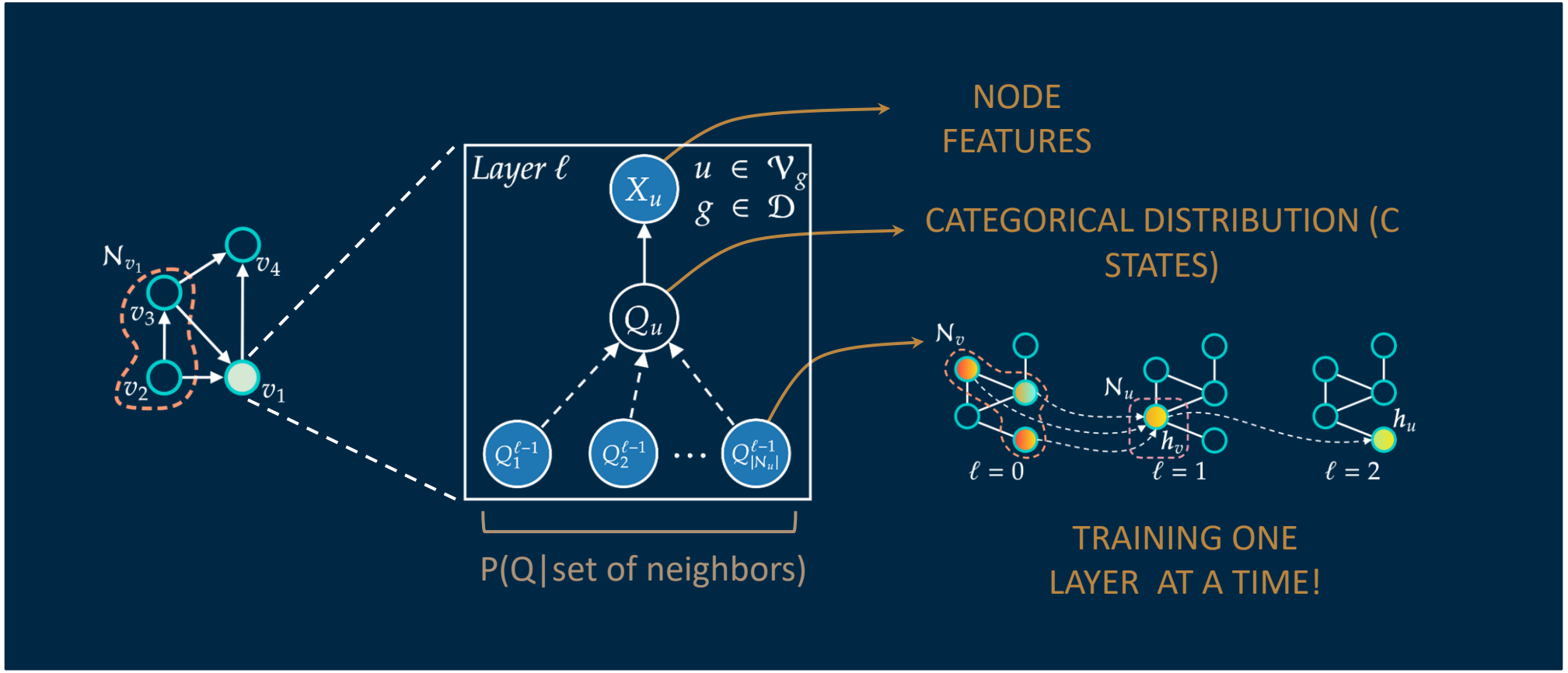
Unsupervised Graph Embeddings

- Learn unsupervised node and graph embeddings
  - Requiring less supervised labelling
  - Reusing embeddings across multiple tasks
- Mix supervised and unsupervised modules



Bacciu, Errica, Micheli ,
ICML 2018

Contextual Graph Markov Model (CGMM)



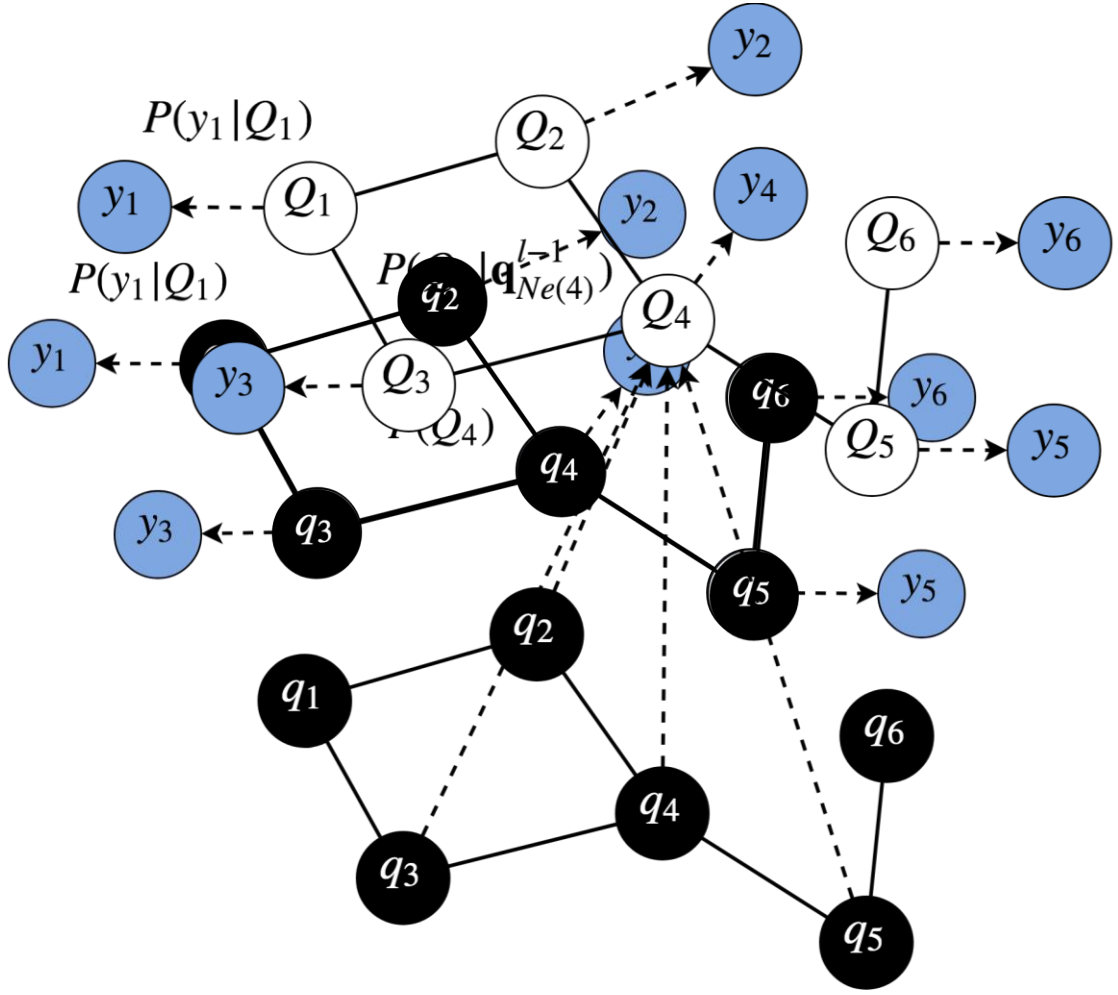
Incremental Construction

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

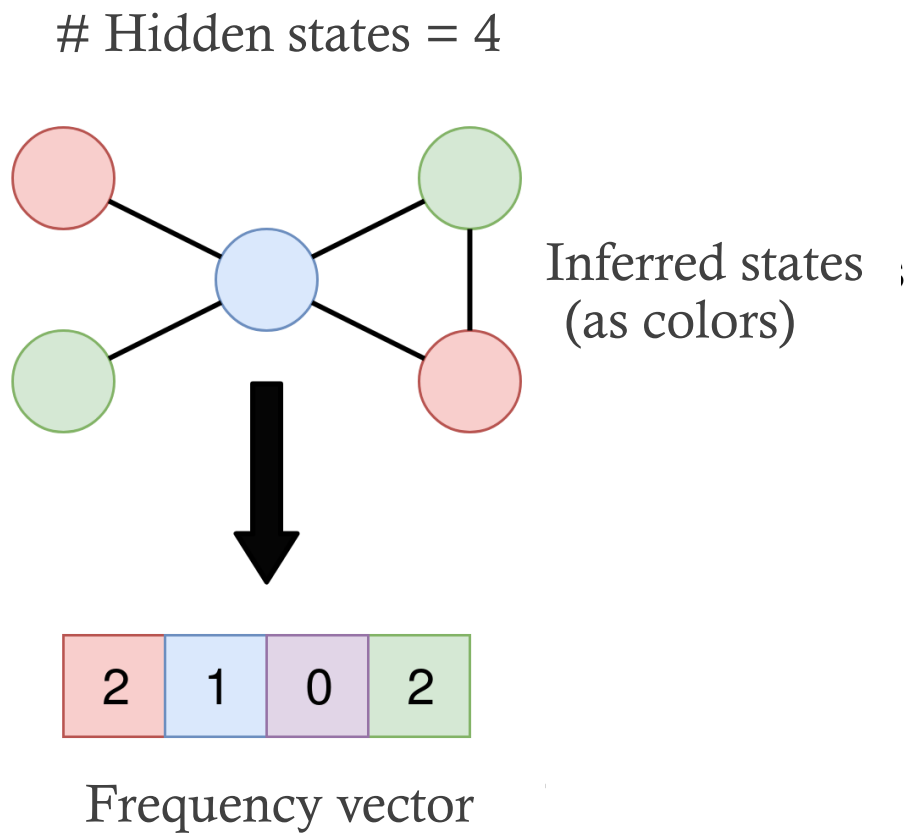


Computing embedding

- 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 Layer Training

A maximum likelihood approach to learning

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

Split by layer

and by arc

$$= \prod_{g \in G} \prod_{u \in \mathrm{g}} \sum_{i}^{C} P^l(y_u | Q_u = i) \sum_{\bar{l} \in \bar{L}} P(L_u = \bar{l}) \sum_{a=1}^{A} P^{\bar{l}}(S_u = a) \frac{1}{|\mathcal{N}^{\bar{l}a}(u)|} \sum_{j}^{C} P^{\bar{l}a}(Q_u = i | Q_*^{\bar{l}a} = j) \sum_{v \in \mathcal{N}^{\bar{l}a}(u)} q_v^{\bar{l}}(j)$$

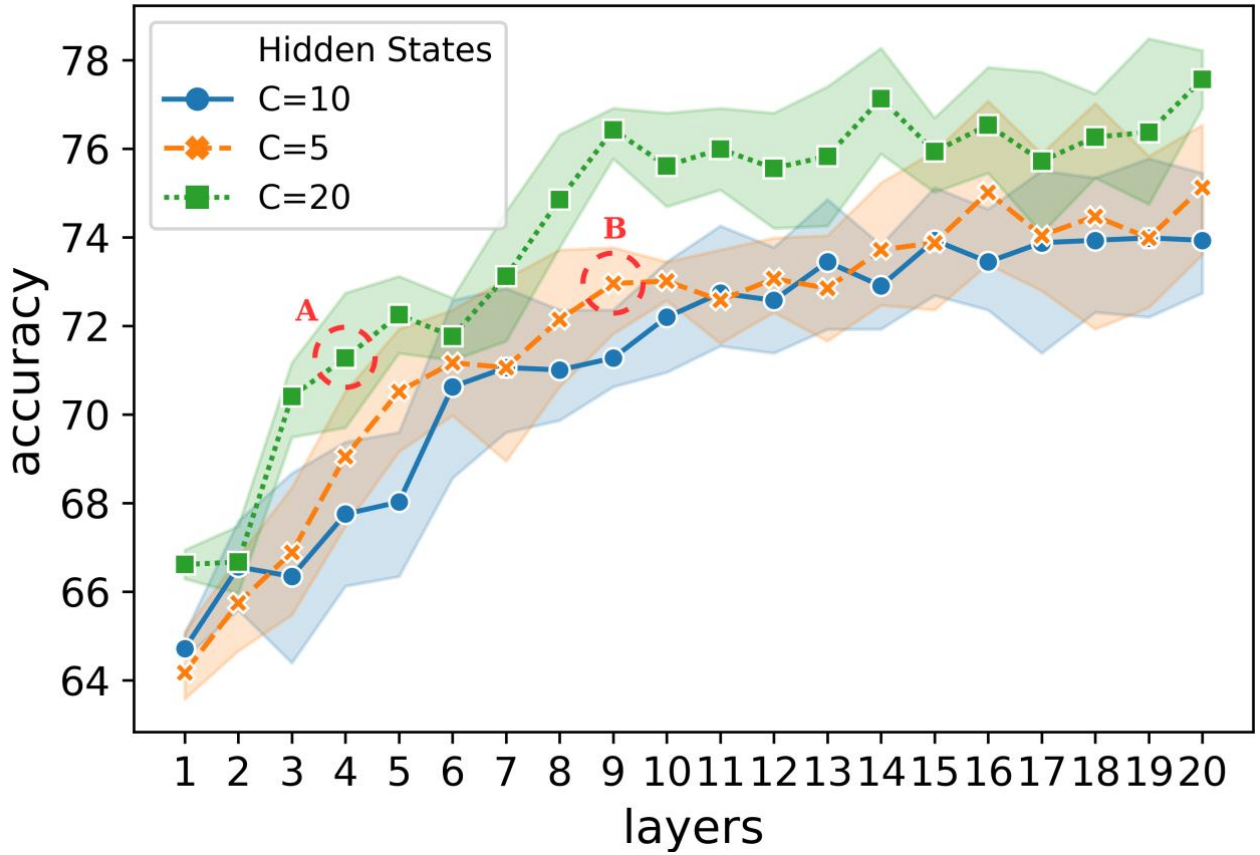
Assumption: i.i.d. graphs

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

Trained by Expectation-Maximization

CGMM – Depth Matters…

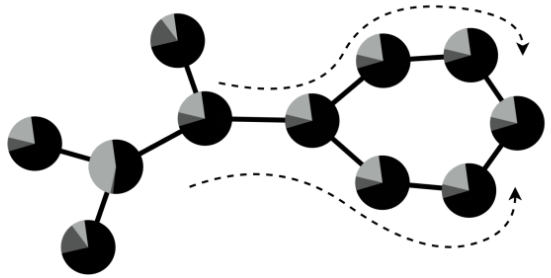
…possibly more than width



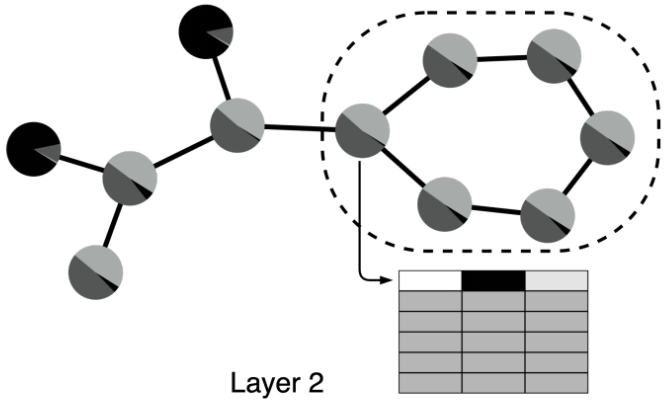
Bacciu, Errica, Micheli , JMLR 2020

Interpreting CGMM

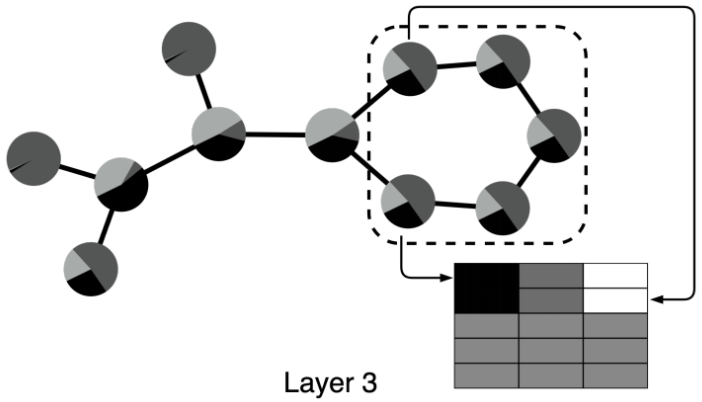
Thanks to the probabilistic approach



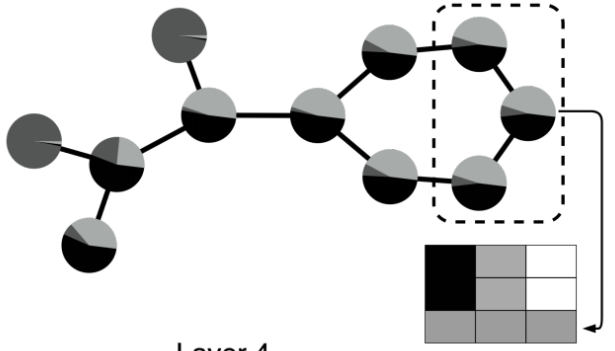
Layer 1



Layer 2



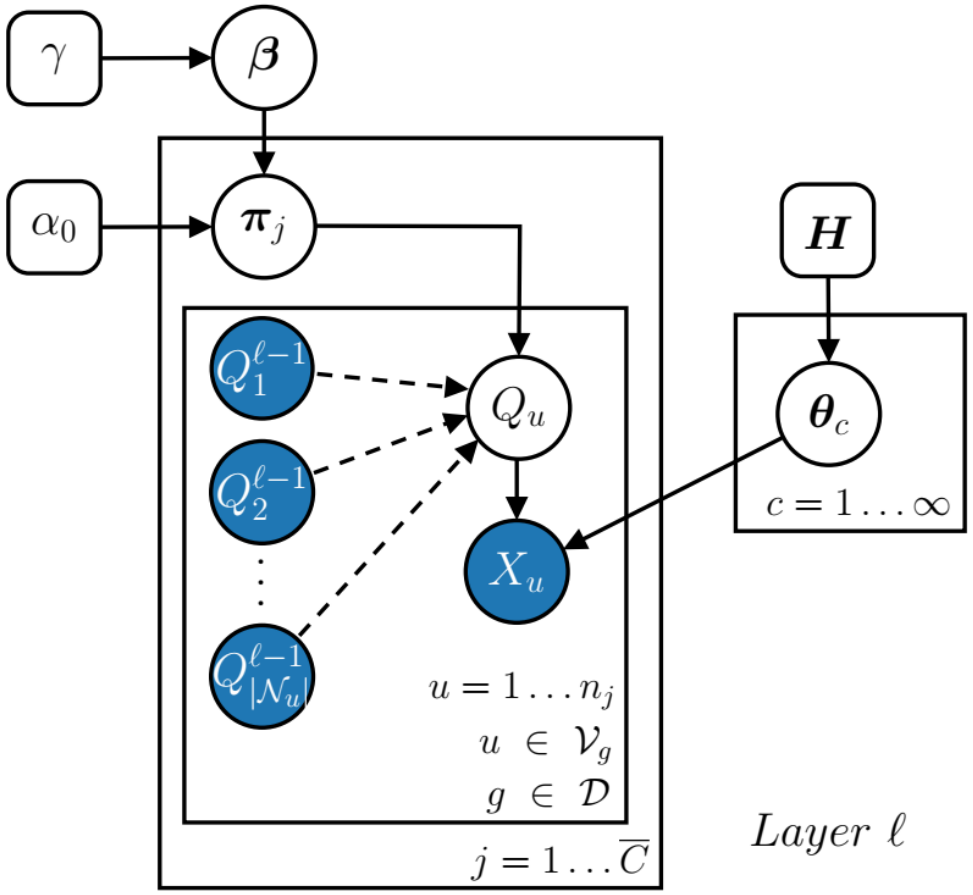
Layer 3



Layer 4

Bacciu, Errica, Micheli , JMLR 2020

To infinity and beyond



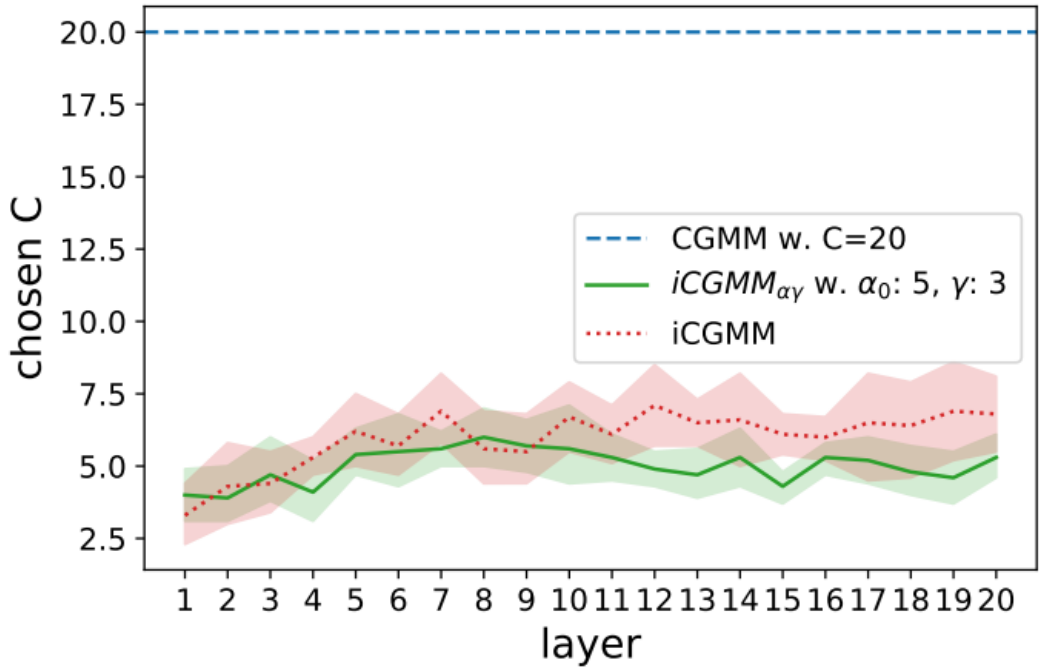
The Infinite CGMM

- Hierarchical Dirichlet process to sample (potentially) infinitely many hidden states
- Automatically learn the size of node embedding space from data
- Choice of observations' groups determined by neighbors' states
- Batch version for larger datasets

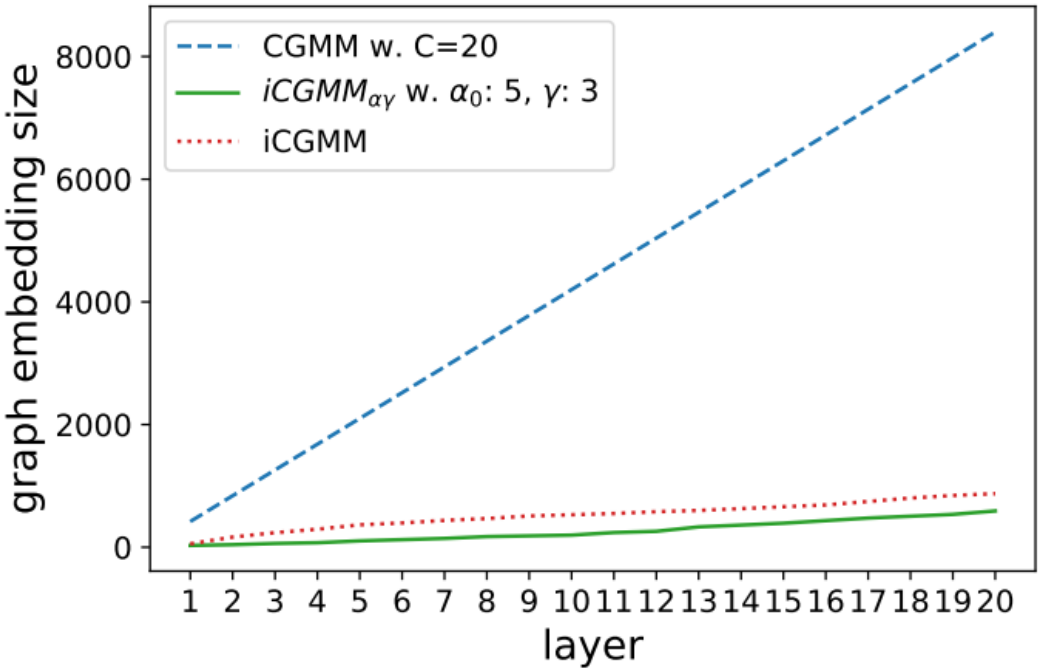
Castellana, Errica, Bacciu, Micheli , ICML 2022

ICGMM – Finer grained control on hidden space

CHOSEN STATES PER LAYER



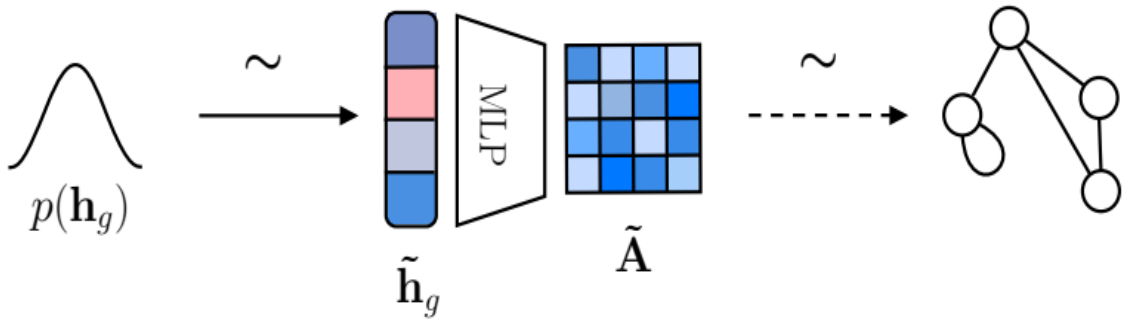
CUMULATIVE GRAPH EMBEDDING SIZE



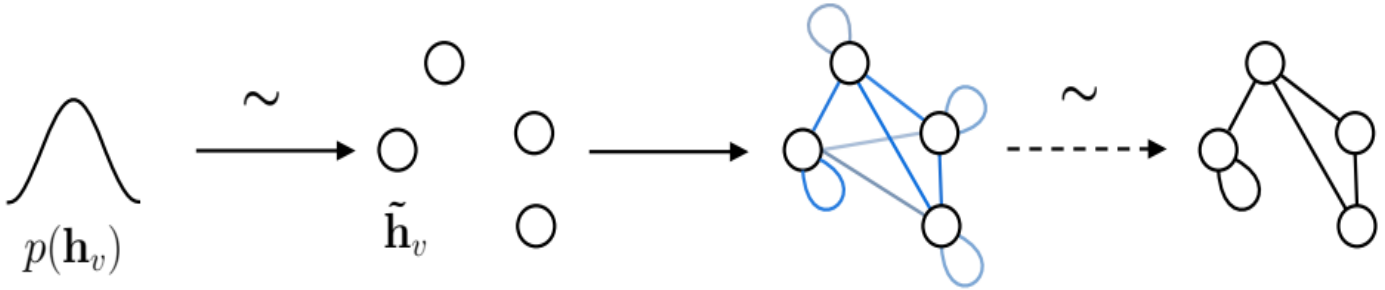
Deep Generative Models for Graphs

Graph Generation

Generate a prediction that is itself a graph

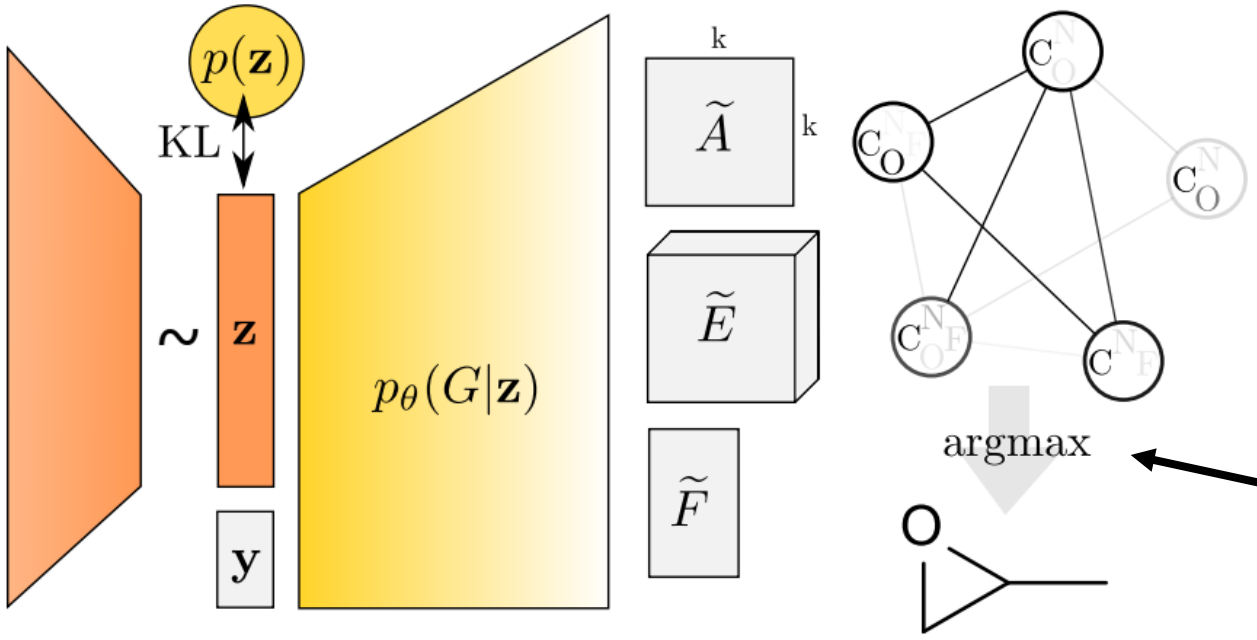


Adjacency-based



Auto-regressive

Graph Variational Autoencoder

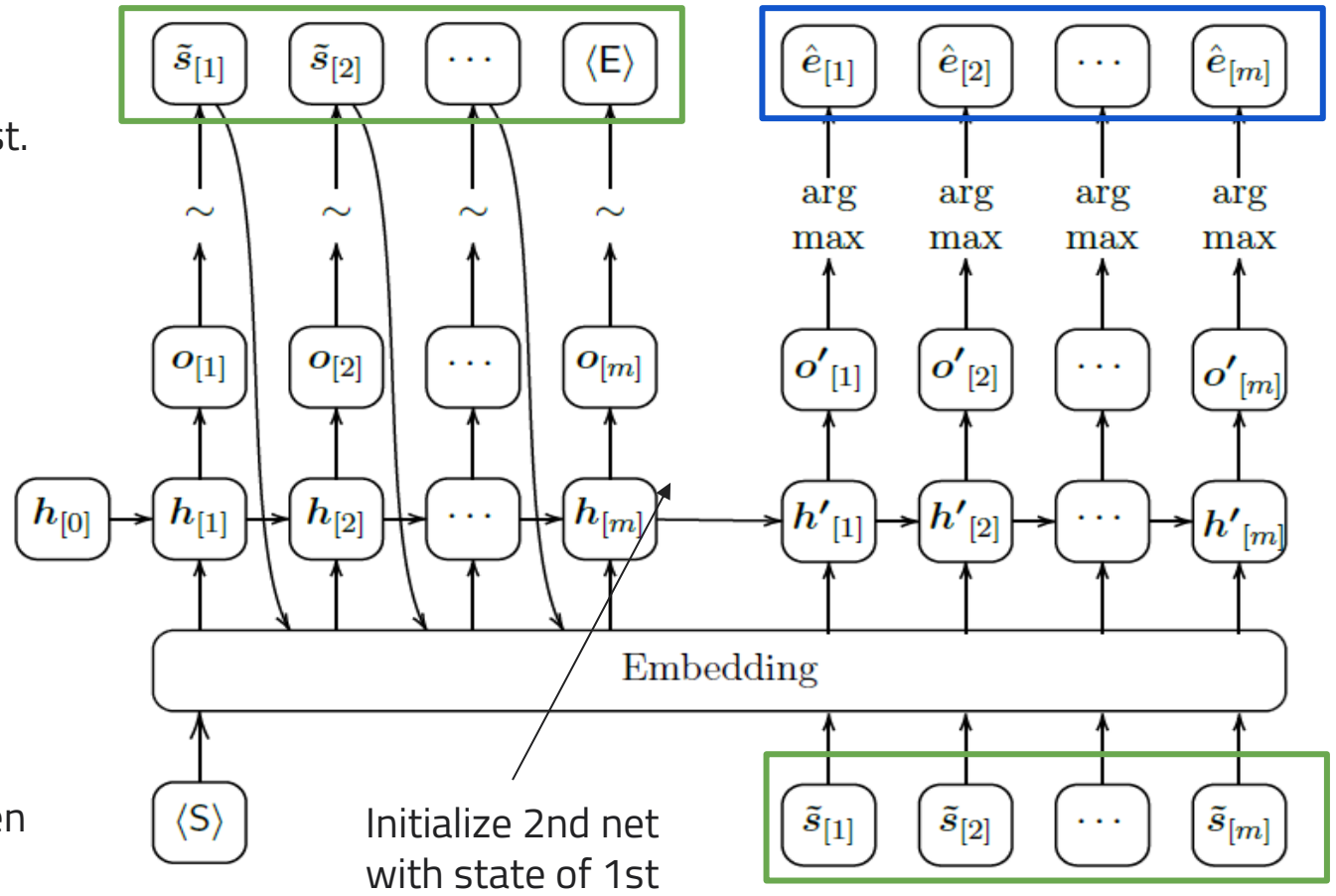


GraphVAE generates adjacency matrix up to k vertices along with the relevant edge/node features (for molecular data)

Argmax a.k.a. sampling ⟹ non-differentiable

Simonovsky, Komodakis, ICLR-WS 2018

Language-Based Graph Generation



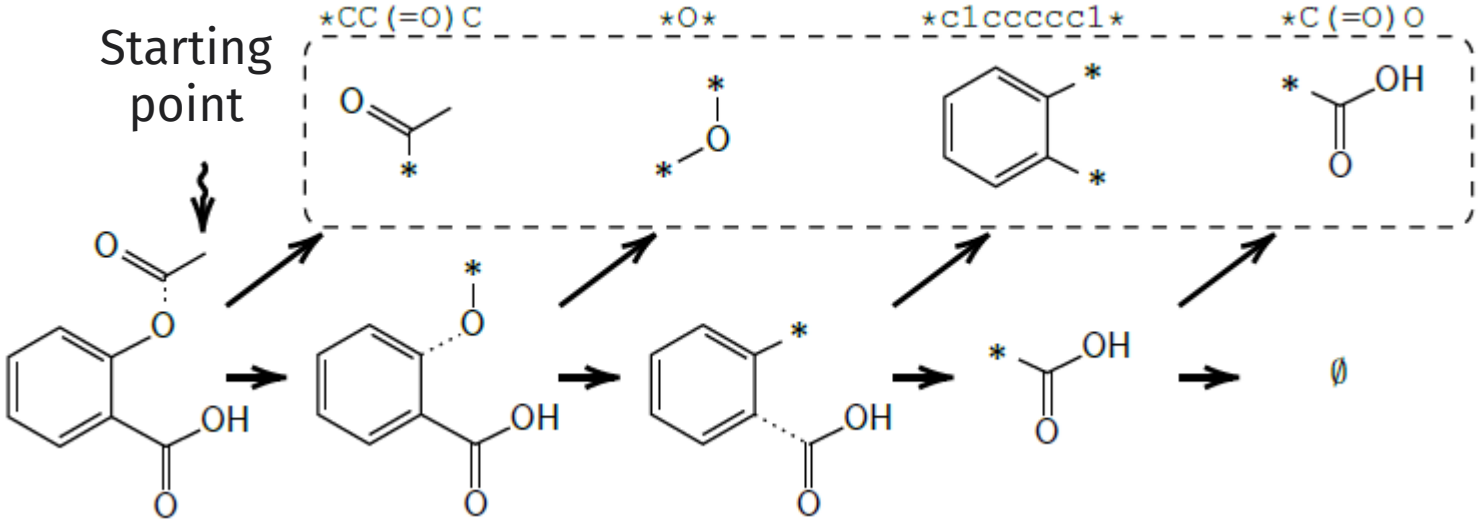
Generate a graph autoregressively: node-by-node and edge-by-edge

Bacciu, Micheli, Podda, Neurocomputing 2020

Generate Molecules by Fragmentation

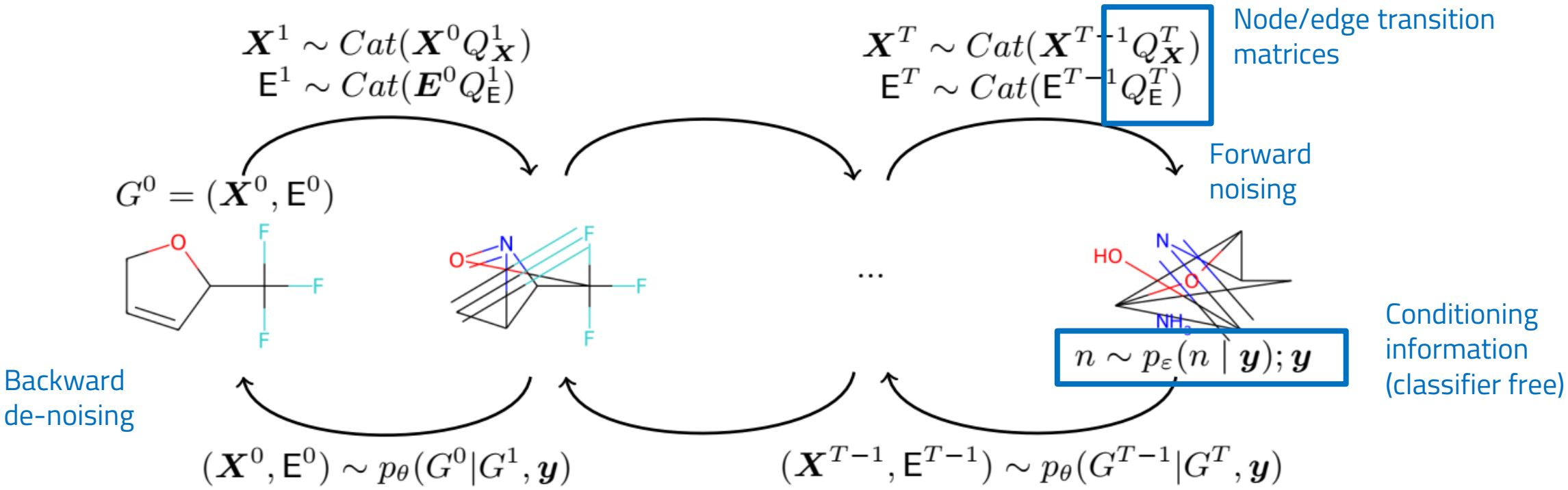
- Molecule is scanned in SMILES order
- Find first breakable bond
- Break the molecule at that bond, set aside leftmost fragment
- Proceed recursively on rightmost fragment

Podda et al, AISTATS 2020



- Order is deterministic and the molecule can be reconstructed
- Keep a vocabulary of all possible fragments found in a dataset
- Graphs are transformed into fragment sequences

Diffusion Models for Graphs

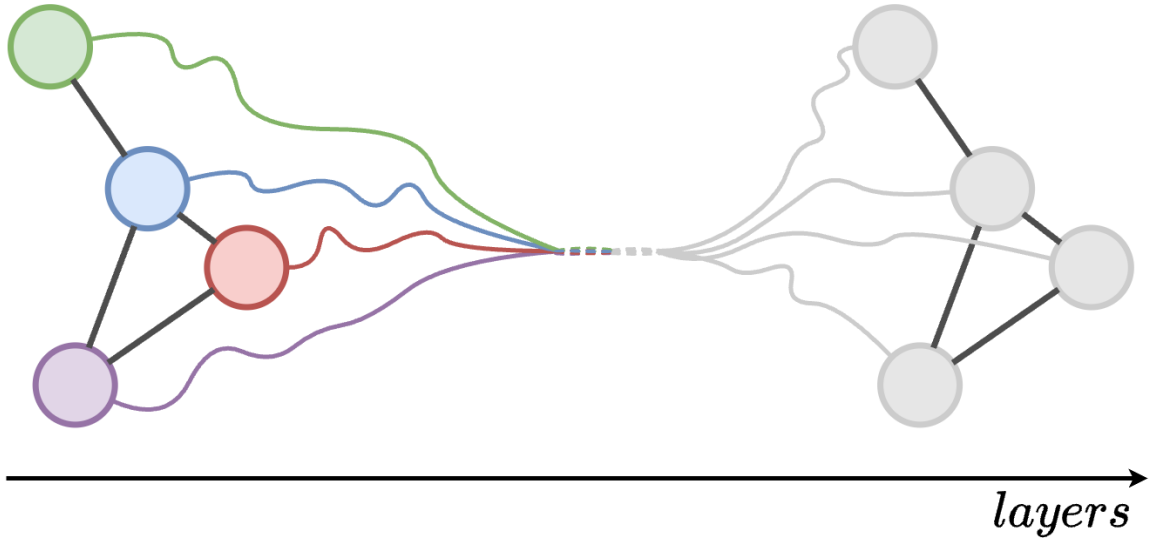


Ninniri, Podda, Bacciu ECML-PKDD 2024

Information Propagation in Graphs

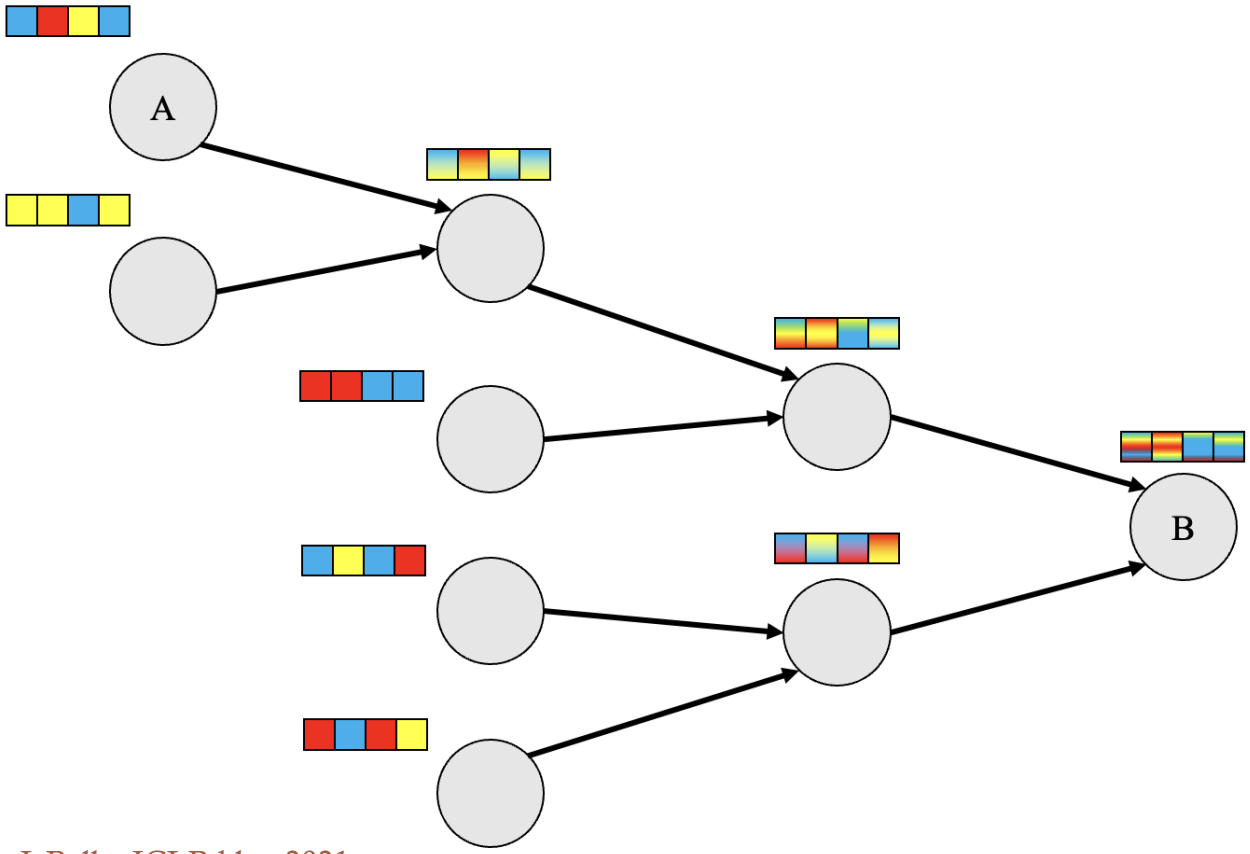
(Catastrophic) Issues when learning node/graph embeddings

Many-layer networks are needed to capture long range node interactions into representative embeddings passing through topological bottlenecks



- That is where our troubles begin
- Under-reaching
- Over-smoothing
- Over-squashing

Oversquashing

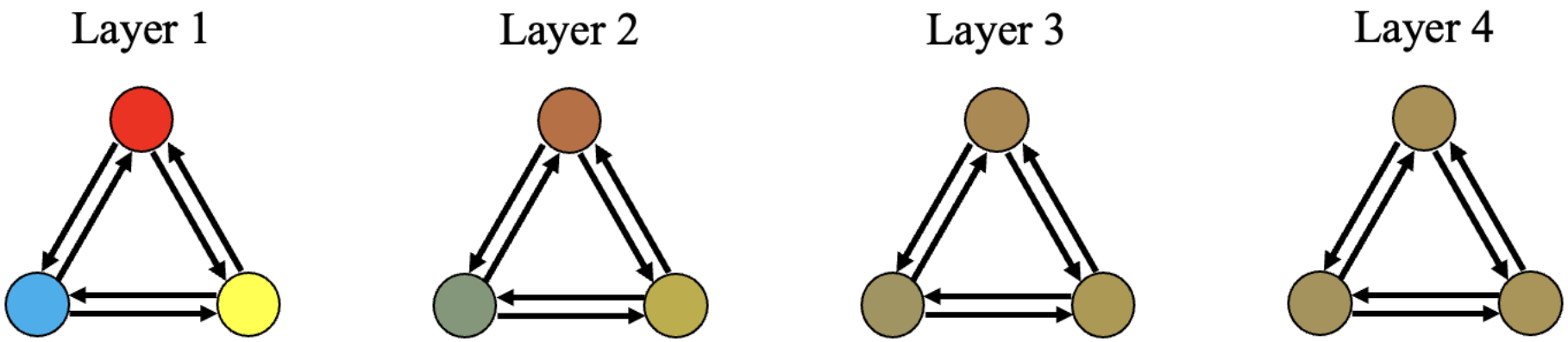


Img: J. Balla, ICLR blog 2021

Occurs when an exponentially-growing amount of information is squashed into a fixed-size vector

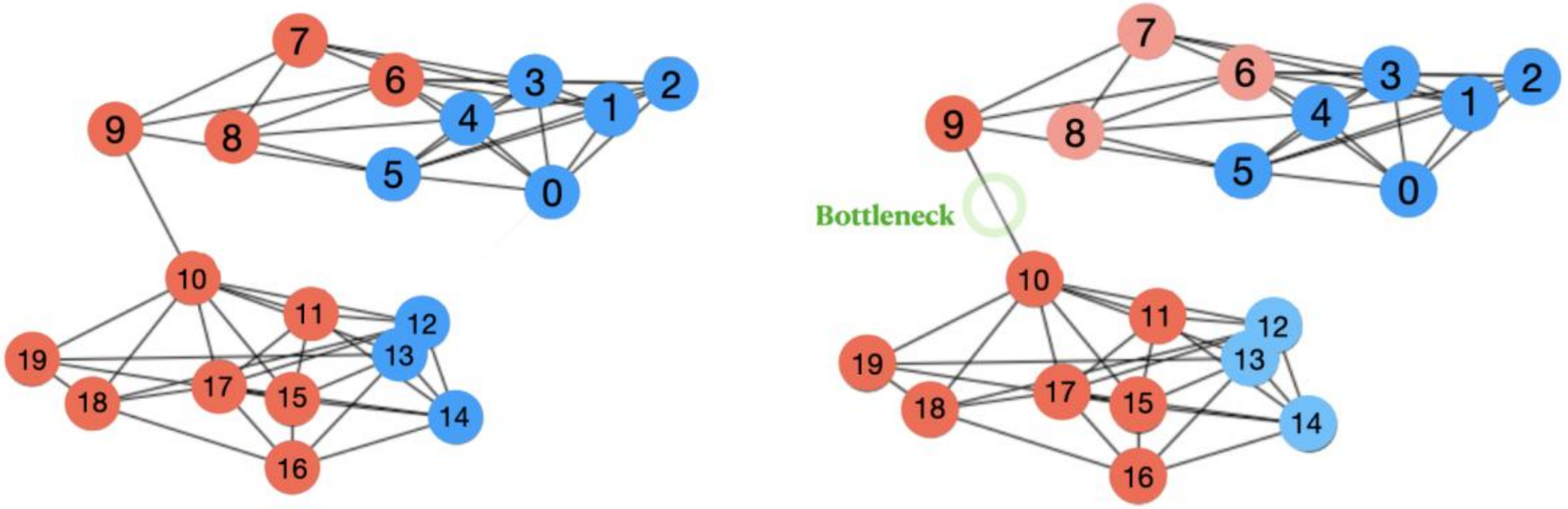
Oversmoothing

Occurs when irrespectively of the propagation length required, the model cannot learn distinctive embeddings to solve the task



Img: J. Balla, ICLR blog 2021

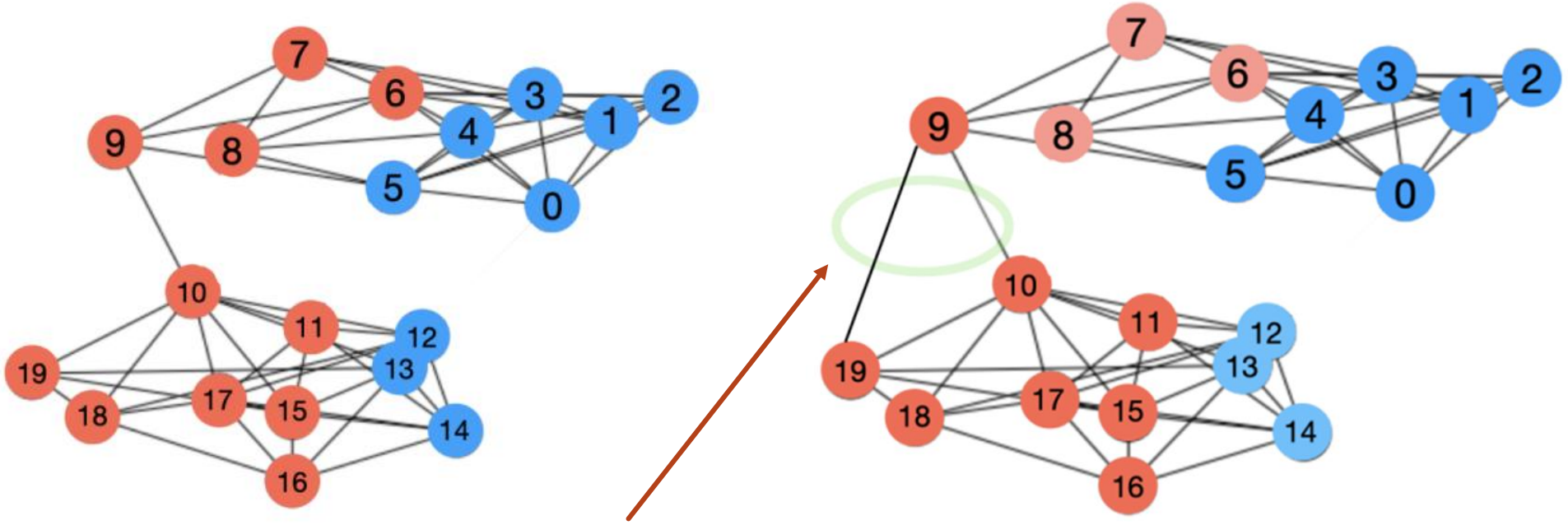
Rewiring Approaches



Topping et al, ICLR 2022

Analysing graph message passing as a diffusion process

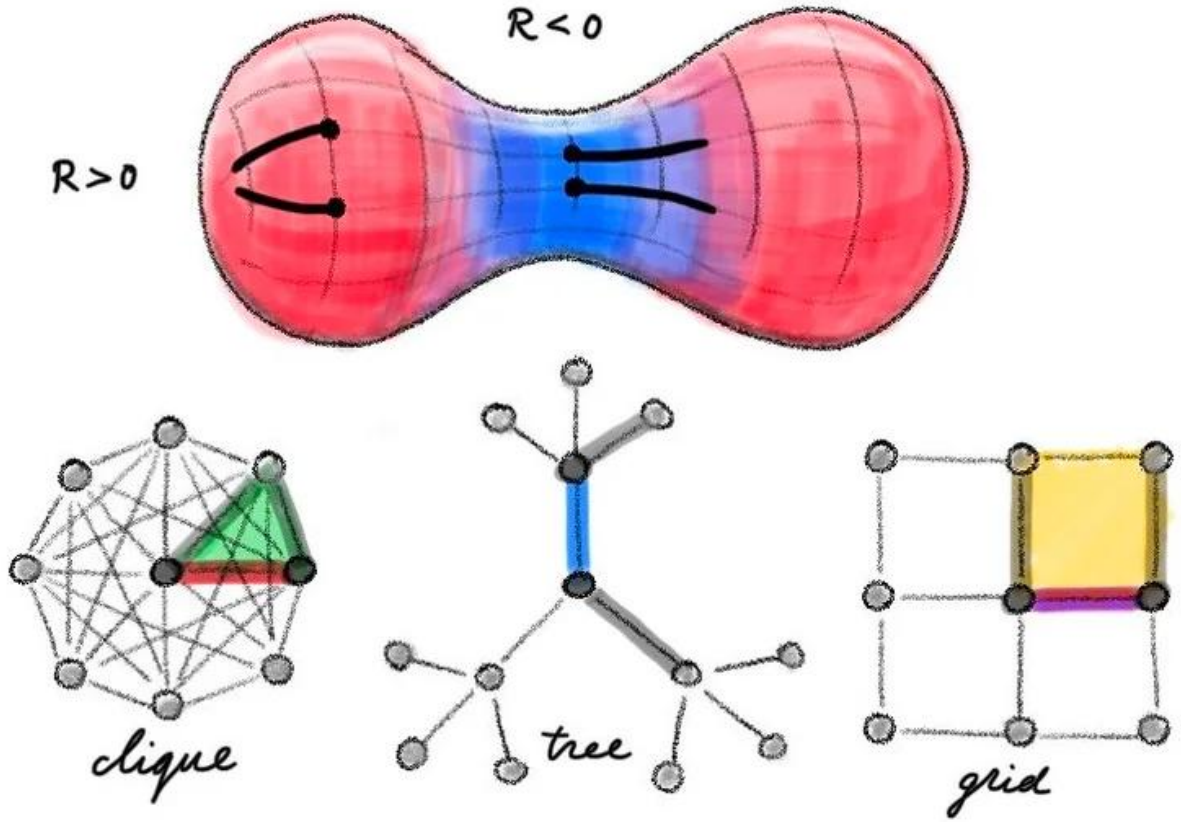
Rewiring Approaches



Reduce the bottleneck (e.g. by a targeted increase in connectivity) to solve over-squashing

Topping et al, ICLR 2022

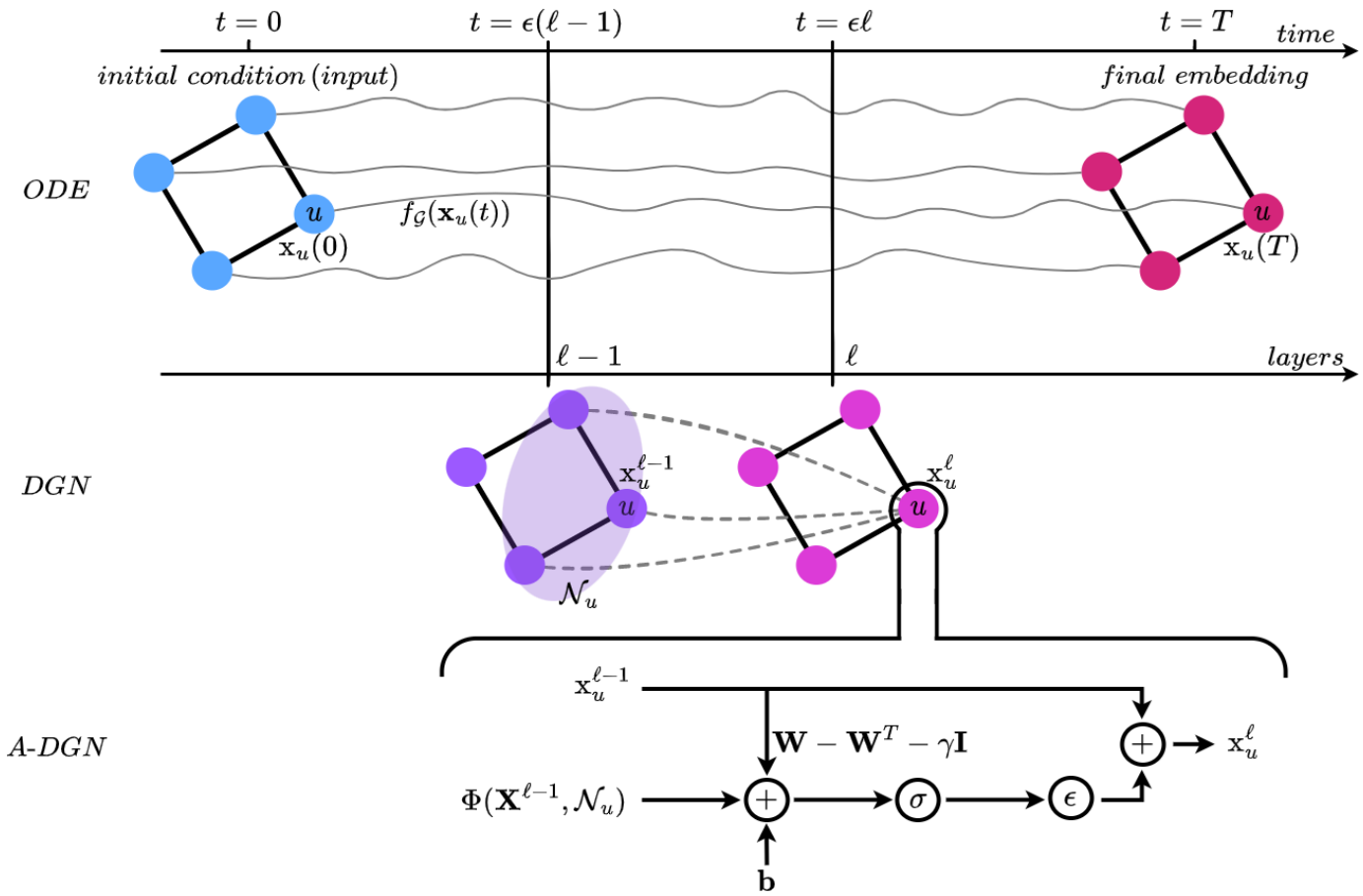
A Topological Perspective



- Connecting message-passing issues with the topological properties (curvature) of graphs
- Negative curvature R maybe one of the causes of oversquashing

Topping et al, ICLR 2022

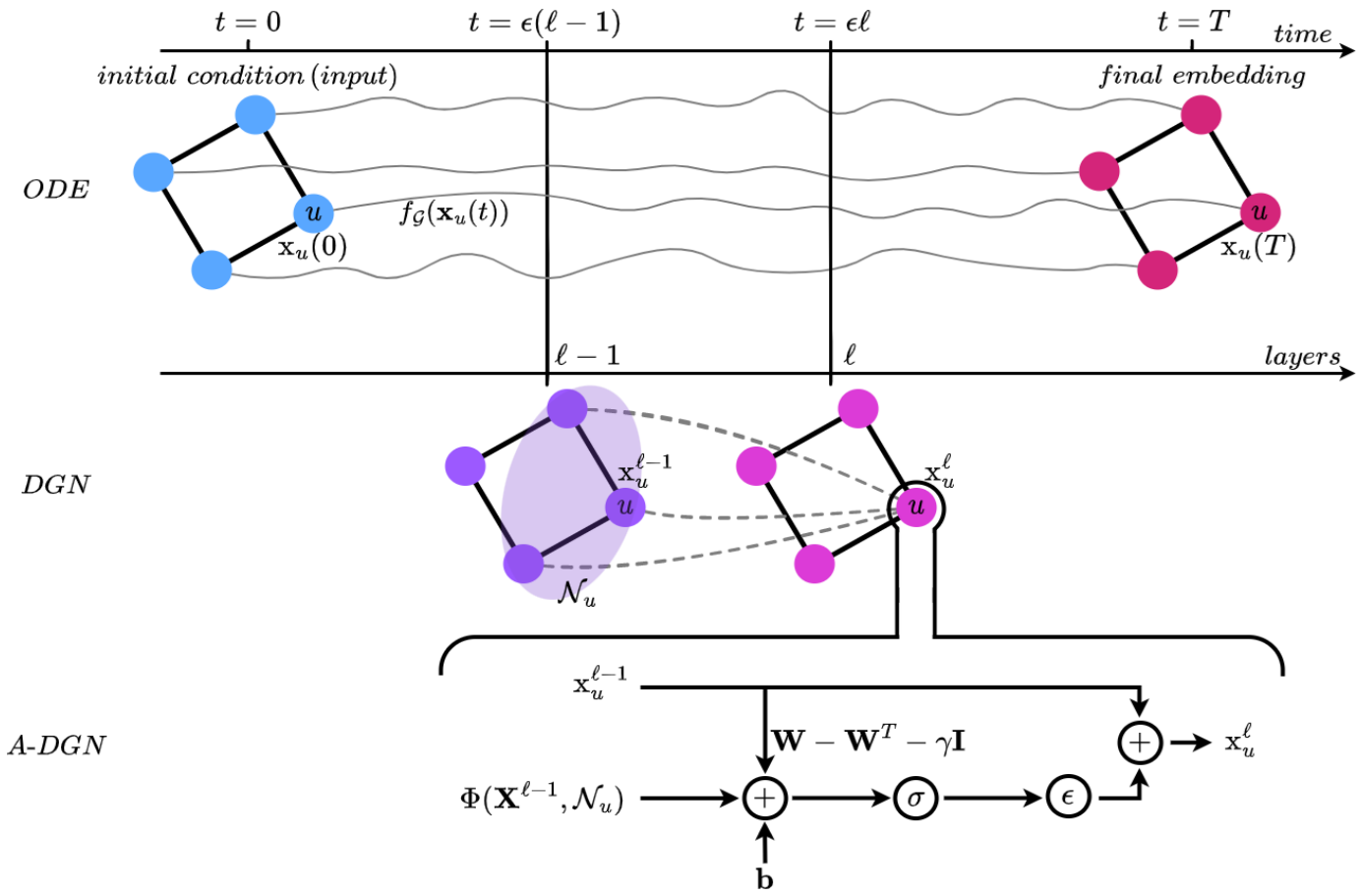
A Dynamical Systems View on Deep Graph Networks



- Node message passing can also be seen as a discretization of a continuous dynamical process

$$\begin{cases} \dfrac{\partial \mathbf{x}_u(t)}{\partial t} = f_{\mathcal{G}}(\mathbf{x}_u(t)) & t \in [0, T], \\ \mathbf{x}_u(0) = \mathbf{x}_u^0 \in \mathbb{R}^d \end{cases}$$

A Dynamical Systems View on Deep Graph Networks



- Node message passing can also be seen as a discretization of a continuous dynamical process
- The graph neural network has as many layers as the length of the unfolded ODE
- Neural (Graph) ODE

Non-Dissipative Propagation – Addressing the Problem through the Dynamical System

Leverage the ODE formulation of DGNs to optimize forward and backward message propagation

$$\frac{d\mathbf{x}_u(t)}{dt} = \sigma\left(\mathbf{W}_t\mathbf{x}_u(t) + \Phi(\{\mathbf{x}_v(t)\}_{v\in\mathcal{N}_u}) + \mathbf{b}_t\right)$$

Node-wise propagation

Neighborhood aggregator (any standard MPNN)

Chase optimal propagation by enforcing a stable dynamics + non-dissipation of the input over time by looking into the properties of Jacobian $\mathbf{J}$

$$\Longrightarrow \forall i: \left(Re\left(\lambda_i(\boldsymbol{J}(t))\right)\right) \approx 0$$

Haber & Ruthotto, 2017
Gravina et al, ICLR 2023

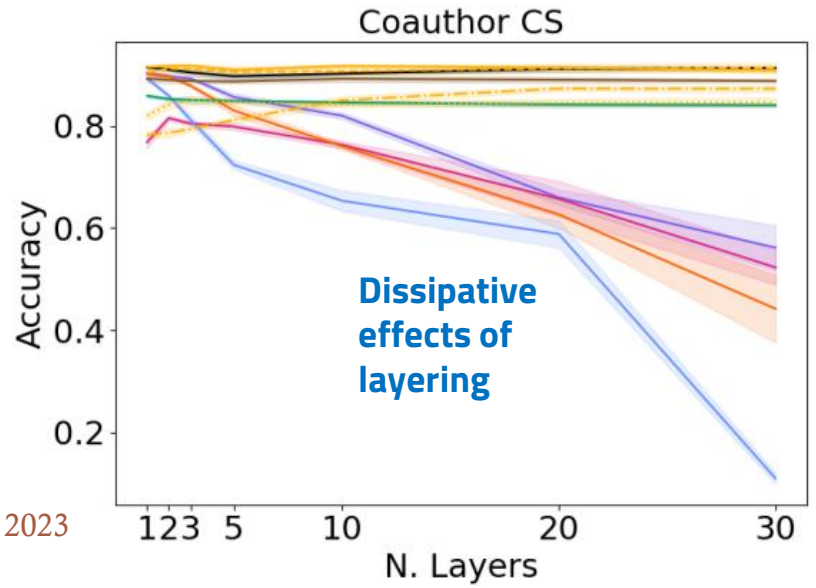
Non-Dissipative Propagation by Anti-Symmetry

Anti-symmetric weight matrix with a Euler discretization of the Neural ODE

$$\mathbf{x}_u^{\ell} = \mathbf{x}_u^{\ell-1} + \epsilon\sigma\left((\mathbf{W} - \mathbf{W}^T - \gamma\mathbf{I})\mathbf{x}_u^{\ell-1} + \Phi(\mathbf{X}^{\ell-1}, \mathcal{N}_u) + \mathbf{b}\right)$$

Forward Euler discretization of Graph ODE

Anti-symmetric weight matrix allowing stable and non-dissipative behavior of the ODE (eigenvalues of the Jacobian are all imaginary)



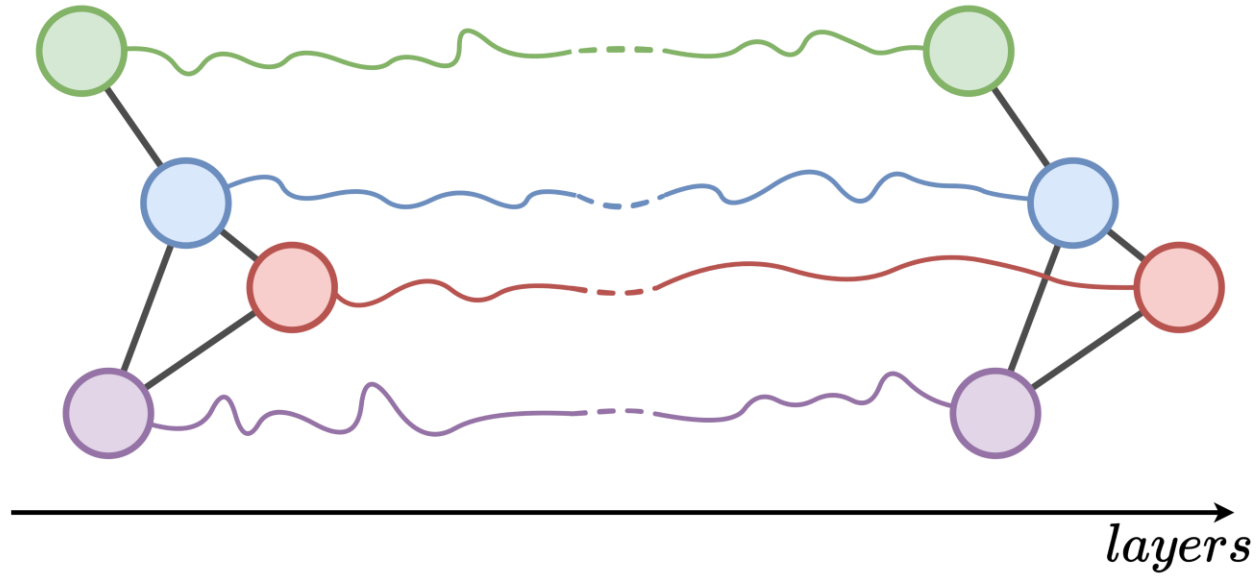
Non-dissipative & rewiring

Gravina et al, ICLR 2023

Local Vs Global Non-Dissipation

So far we have achieved local conservation

$$\left\| \frac{\partial \boldsymbol{x}_u(t)}{\partial \boldsymbol{x}_u(0)} \right\| \approx c$$

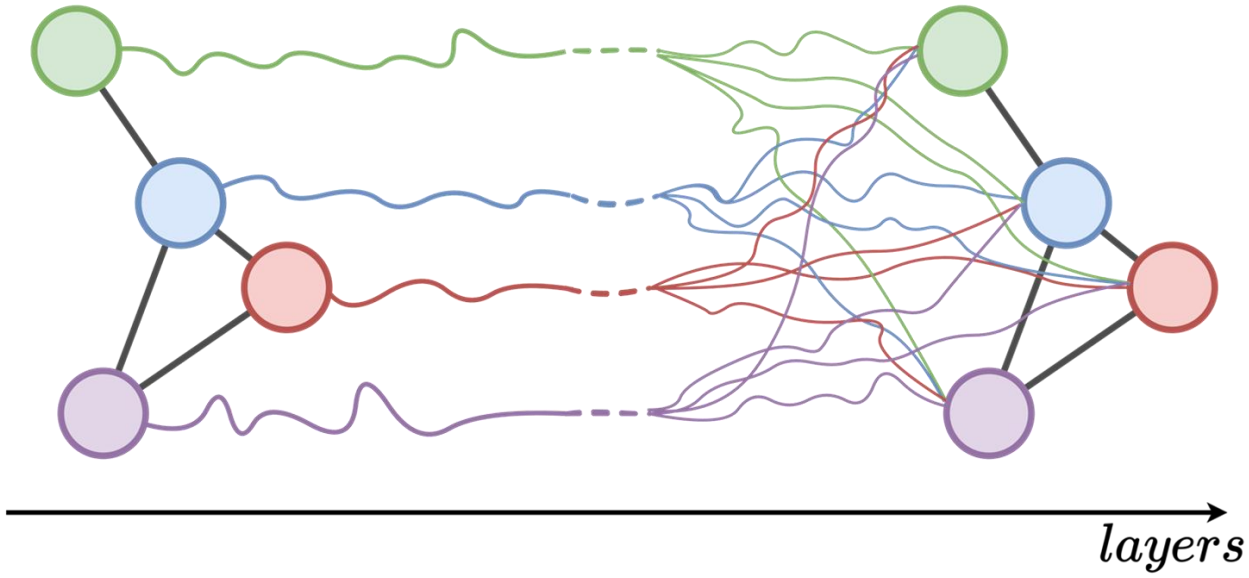


Gravina et al, ICLR 2023

Local Vs Global Non-Dissipation

But we want to achieve it globally

$$\left\| \frac{\partial vec(\boldsymbol{X}(t))}{\partial vec(\boldsymbol{X}(0))} \right\| \approx c$$



Gravina et al, Arxiv 2024

SWAN – Space and Weight Antisymmetry Network

$$\frac{d\mathbf{x}_u(t)}{dt} = \sigma\Big((\mathbf{W} - \mathbf{W}^\top)\mathbf{x}_u(t) + \Phi(\{\mathbf{x}_v\}_{v\in\mathcal{N}_u}) + \beta\Psi(\{\mathbf{x}_v\}_{v\in\mathcal{N}_u})\Big)$$

Node-wise Antisymmetric propagation

(any) Neighbourhood aggregation

Antisymmetric neighbourhood aggregation

$$\Psi = \sum_{v\in\mathcal{N}_u} (\tilde{\mathbf{A}}_{uv} - \tilde{\mathbf{A}}_{vu})(\mathbf{Z} + \mathbf{Z}^\top)\mathbf{x}_v(t)$$

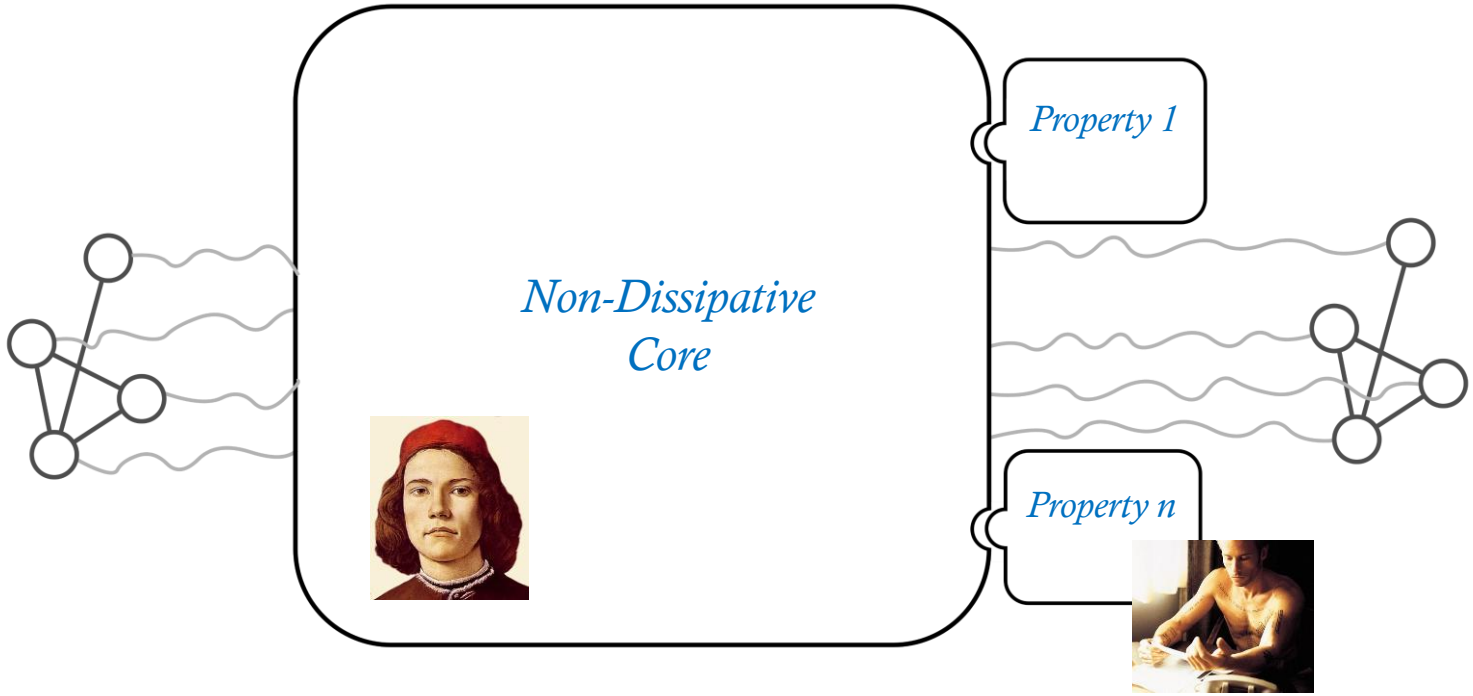
learnable weights

pre-defined/learned neighbourhood aggregation matrices

Gravina et al, AAAI 2025

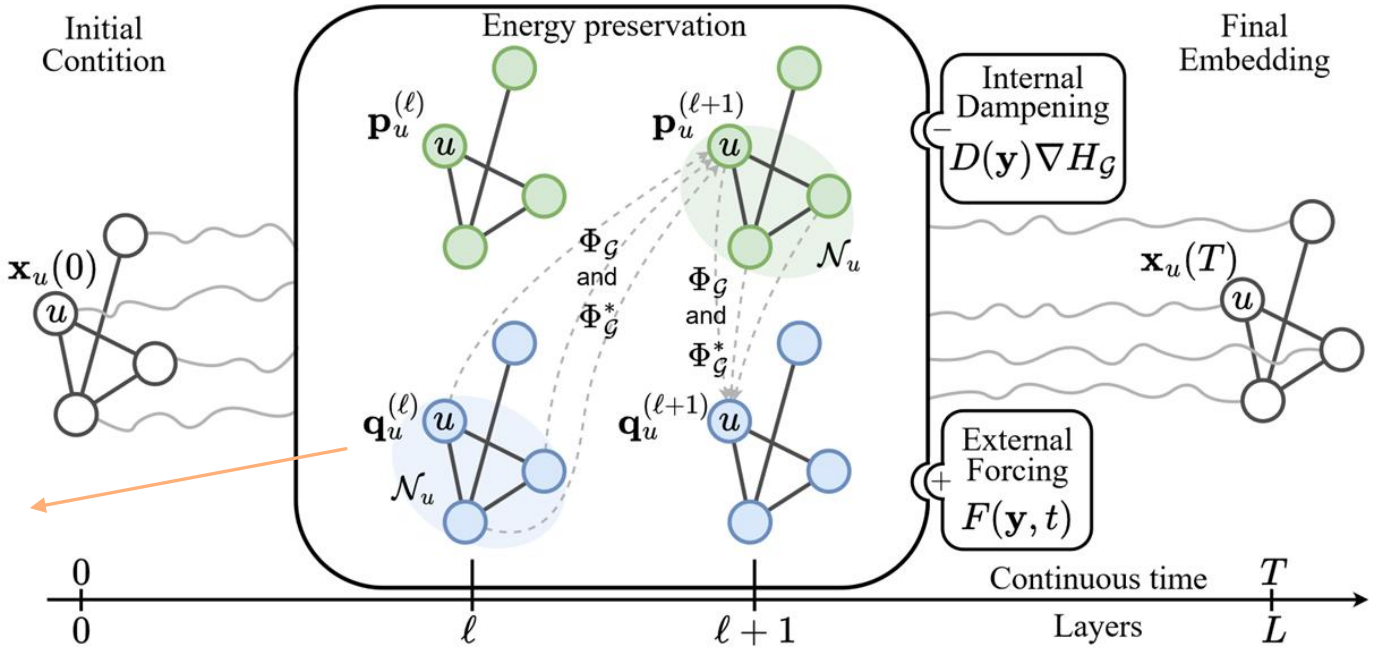
One Further Push

Can we allow to the model to trade between dissipative and non-dissipative behaviors, adaptively and in a principled way?



One Further Push

Can we allow to the model to trade between dissipative and non-dissipative behaviors, adaptively and in a principled way?



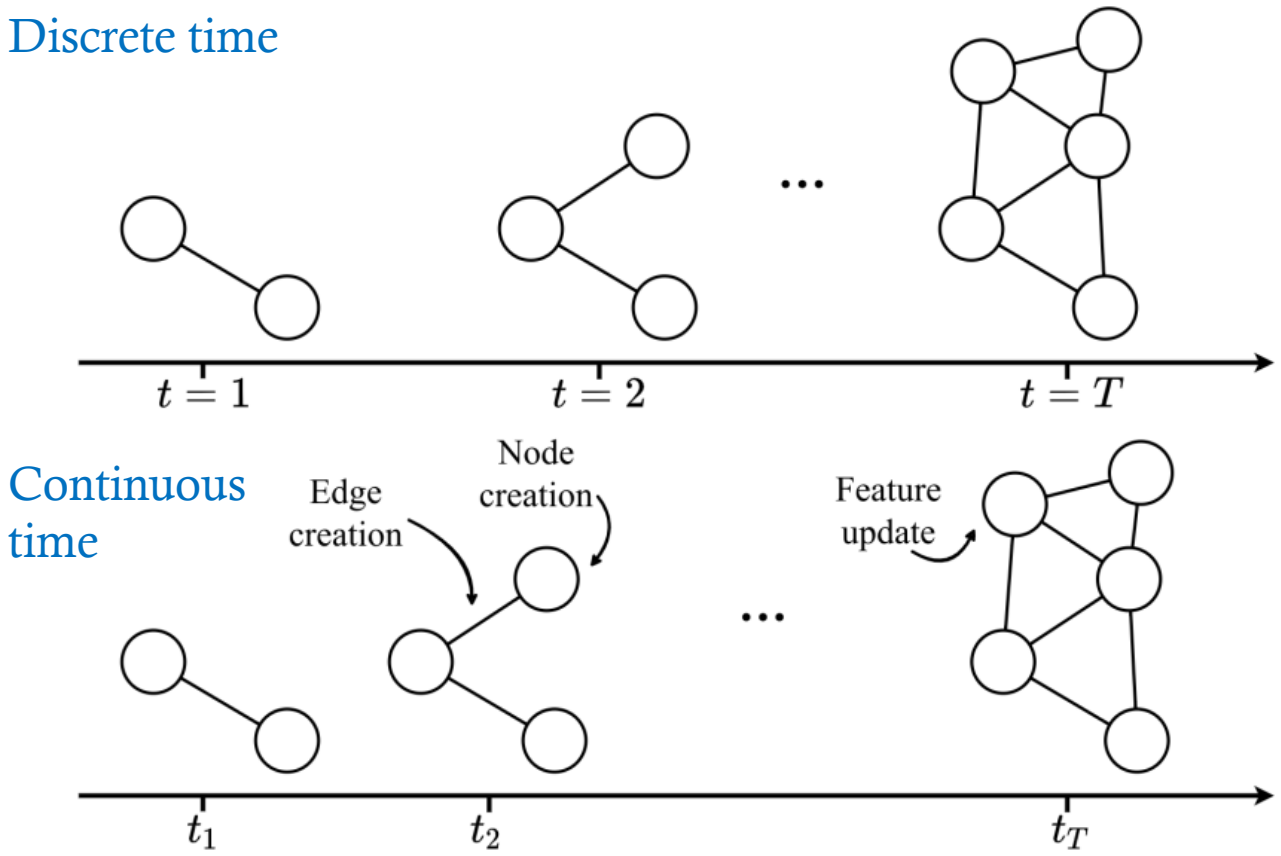
Hamiltonian core evolves global state $\boldsymbol{y}$ preserving energy

Port-Hamiltonian Deep Graph Network (PH-DGN)

Heilig et al, ICLR 2025

Dynamic Graphs

Learning with Dynamic Graphs

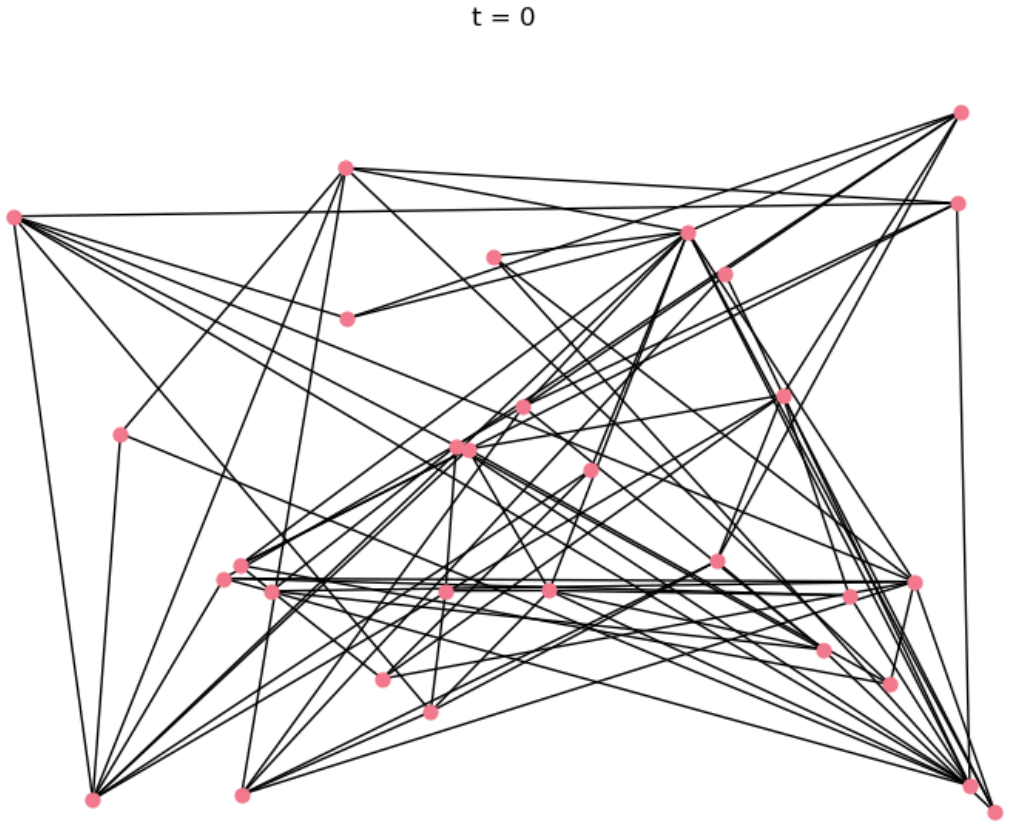


- Graphs evolve with time in feature, connectivity and topology
- Spatio-temporal networks
- Graph streams

R. Trivedi et al ICRL 2019

Gravina & Bacciu, TNNLS 2024 (Survey)

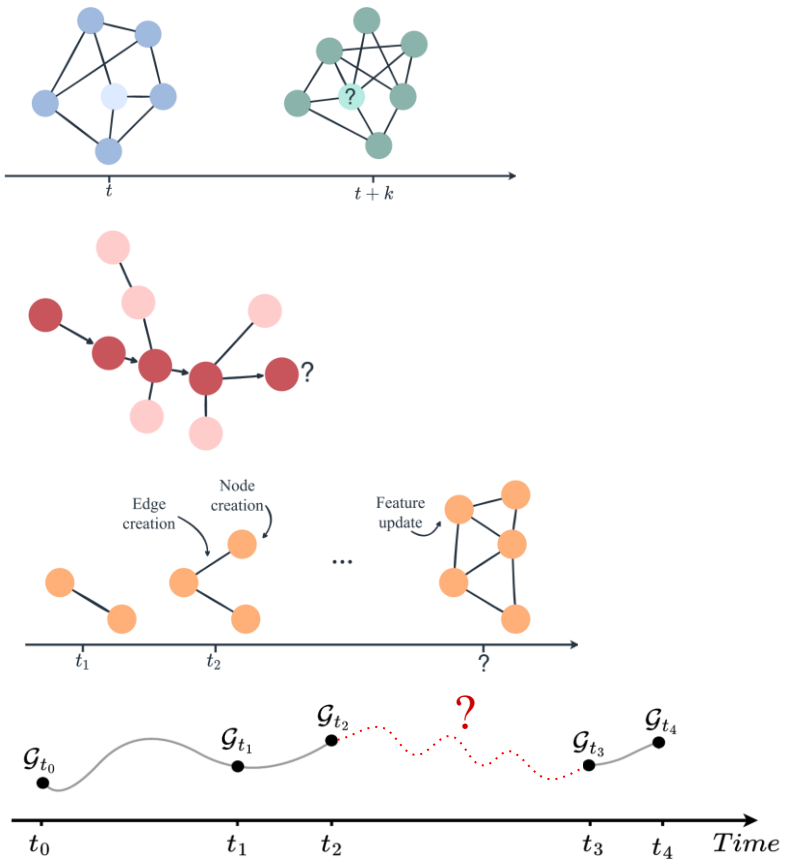
Dynamic Graphs Vs Static DGNs



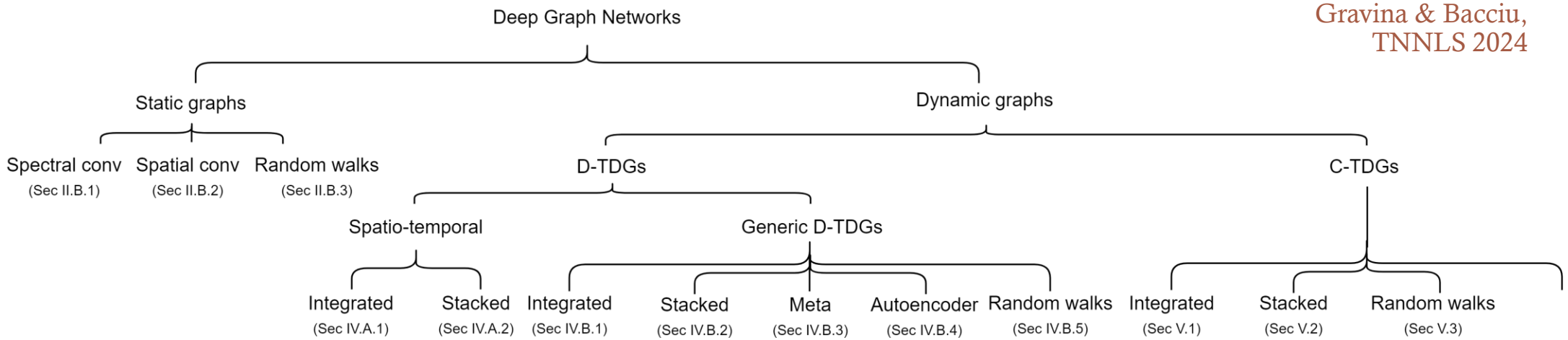
- DGNs cannot be directly applied to all real-life graphs
  - Most real-life graphs are dynamic
  - Majority of DGN approaches assume that the input graph is static
- Ignoring temporal information can make the problem impossible to solve
- Objective: develop methods that are able to exploit both spatial and temporal information

Common Tasks with Dynamic Graphs

- Future link/node prediction
  - Predict at time t + k
- Path classification
  - E.g. predict path congestion
- Event time prediction
  - When an event will occur?
- Imputation



A Taxonomy of Approaches



Gravina & Bacciu, TNNLS 2024

Can again be tackled as a diffusion process using Graph ODEs

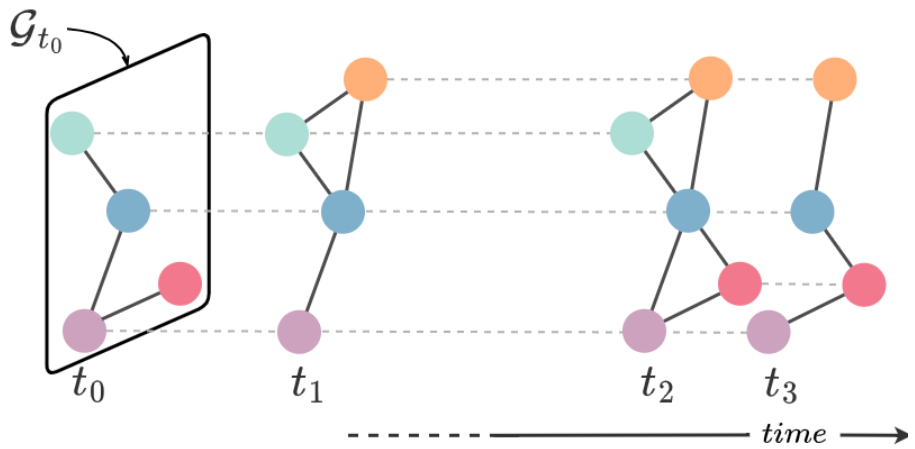
- Spatial and temporal diffusion
- Can be made non-dissipative
- Can naturally handle irregular sampling

A Graph ODE on Continuous Time

A neural ODE that propagates node signals between event occurrences

Can again be solved by forward Euler

$$\mathbf{x}_u^{\ell+1} = \mathbf{x}_u^{\ell} + \epsilon \phi_U \left(\mathbf{x}_u^{\ell}, \mathbf{z}(t_\ell), \rho \left(\{ \phi_M(\mathbf{x}_u^{\ell}, \mathbf{x}_v^{\ell}, \mathbf{e}_{vu}^{\ell}) \}_{v \in \mathcal{N}_u(t_\ell)} \right) \right)$$



Update Aggregate Message

$$\begin{cases} \frac{d\mathbf{x}_u}{dt} = \phi_U \left(\mathbf{x}_u, \mathbf{z}, \rho \left(\{ \phi_M(\mathbf{x}_u, \mathbf{x}_v, \mathbf{e}_{vu}) \}_{v \in \mathcal{N}_u} \right) \right) \\ \mathbf{x}_u(0) = \overline{\mathbf{x}}_{t_{i-1},u} \end{cases}$$

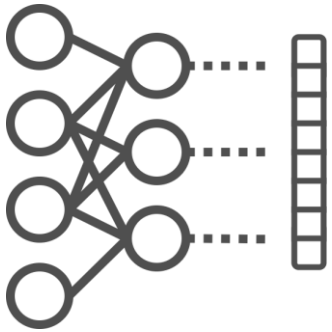
Observed snapshot of u at time t_{i-1}

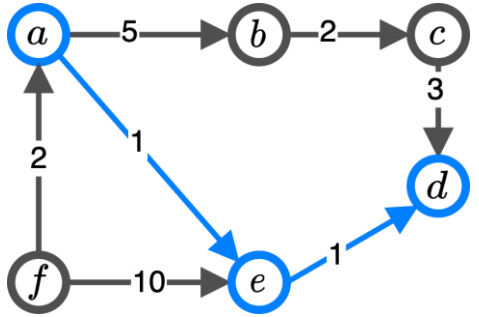
Gravina et al, IJCAI 2024

Integrating Algorithmic Knowledge

Neural Algorithmic Reasoning - Combining algorithms and neural networks

Veličković el al, ICRL 2020



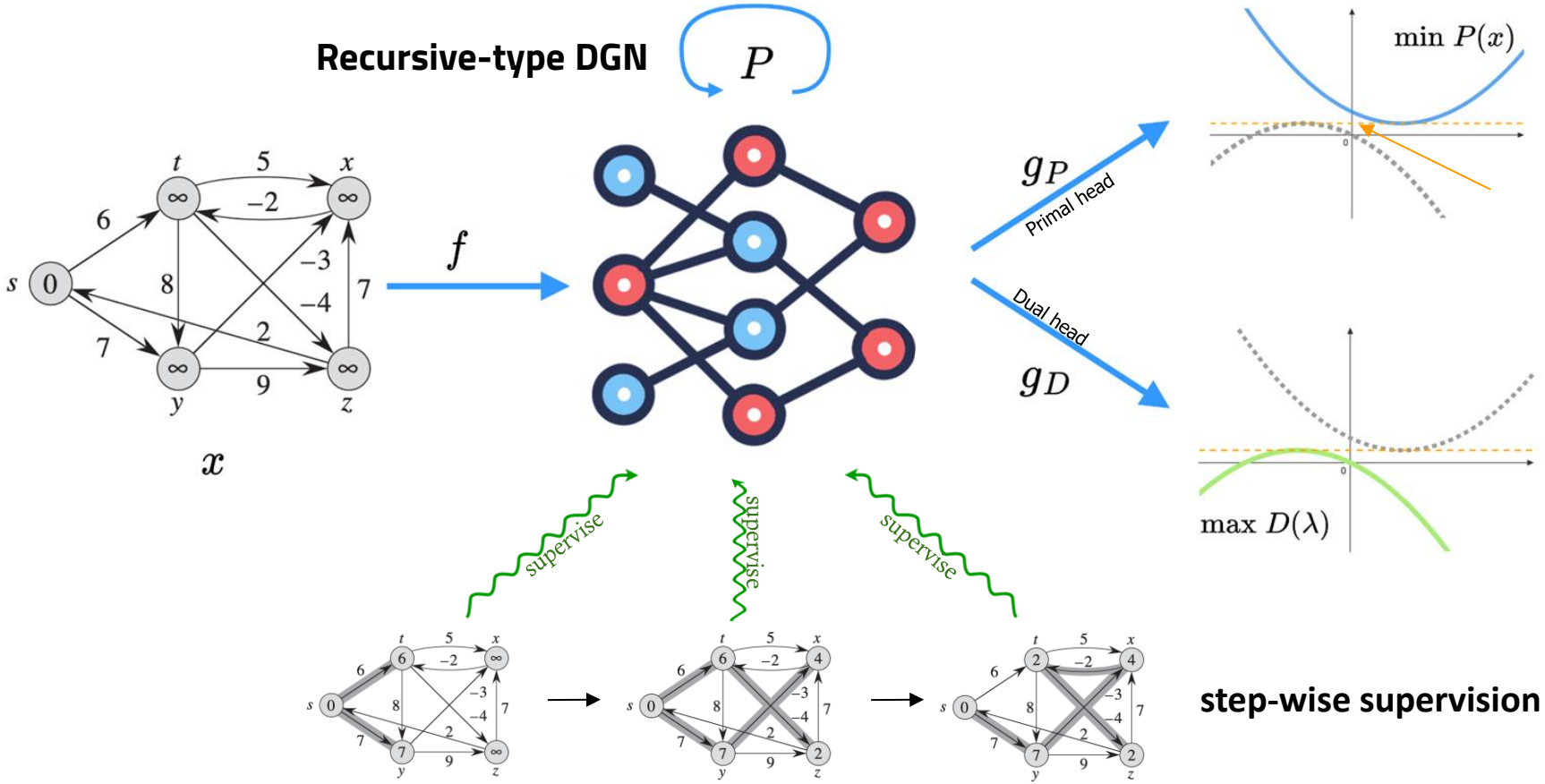


+ Reusable across tasks
+ Executing on noisy conditions
- Sensitive to shift-of-distribution
- No interpretable operations
- Requires lots of data

- Sensitive to task variation
- Input must match pre-conditions
+ Inherent generalisation
+ Interpretable
+ Theoretical guarantees

Can we get the best of both worlds?

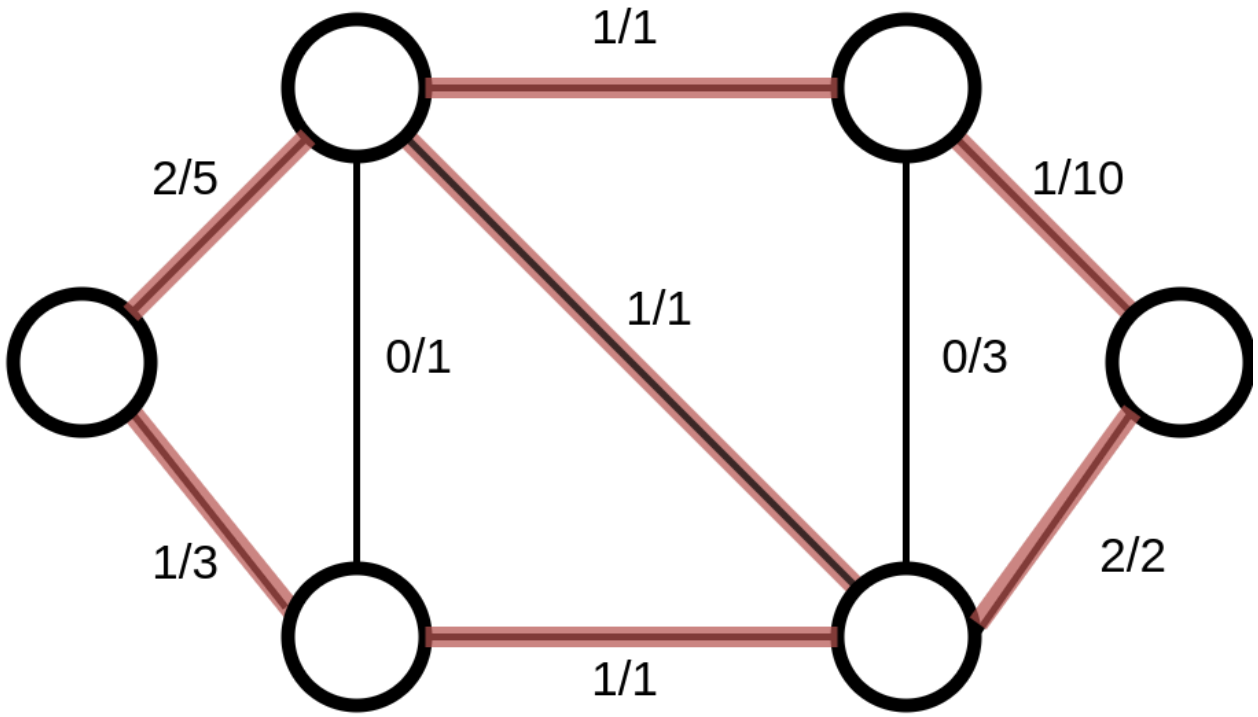
Learning Algorithmic Reasoning on Graphs



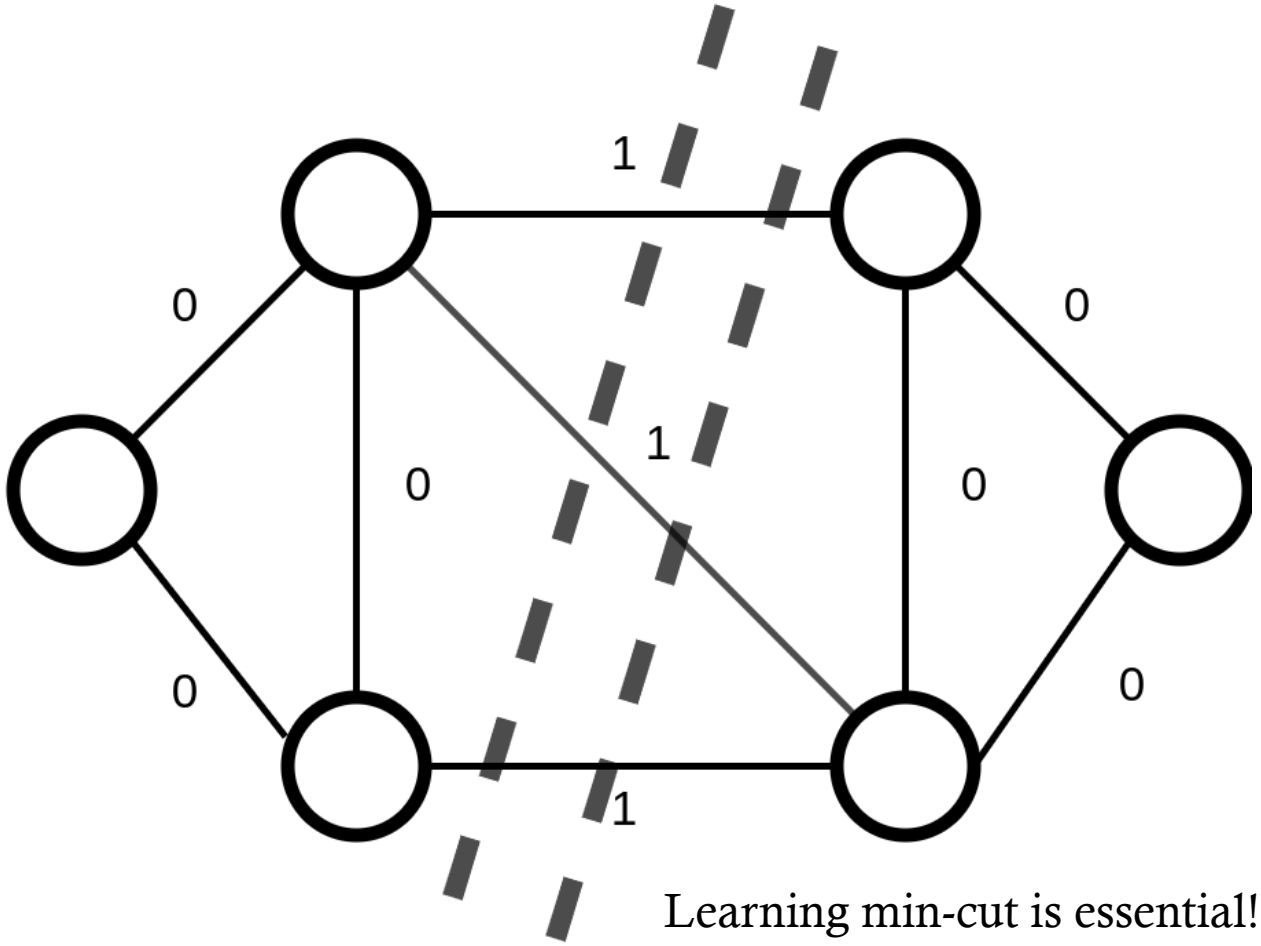
Dual Algorithmic Reasoning
Use knowledge on the structure of the optimization problem

Numeroso, Bacciu, Velickovic, ICLR 2023

Example: Ford-Fulkerson, Max-Flow & Min-Cut



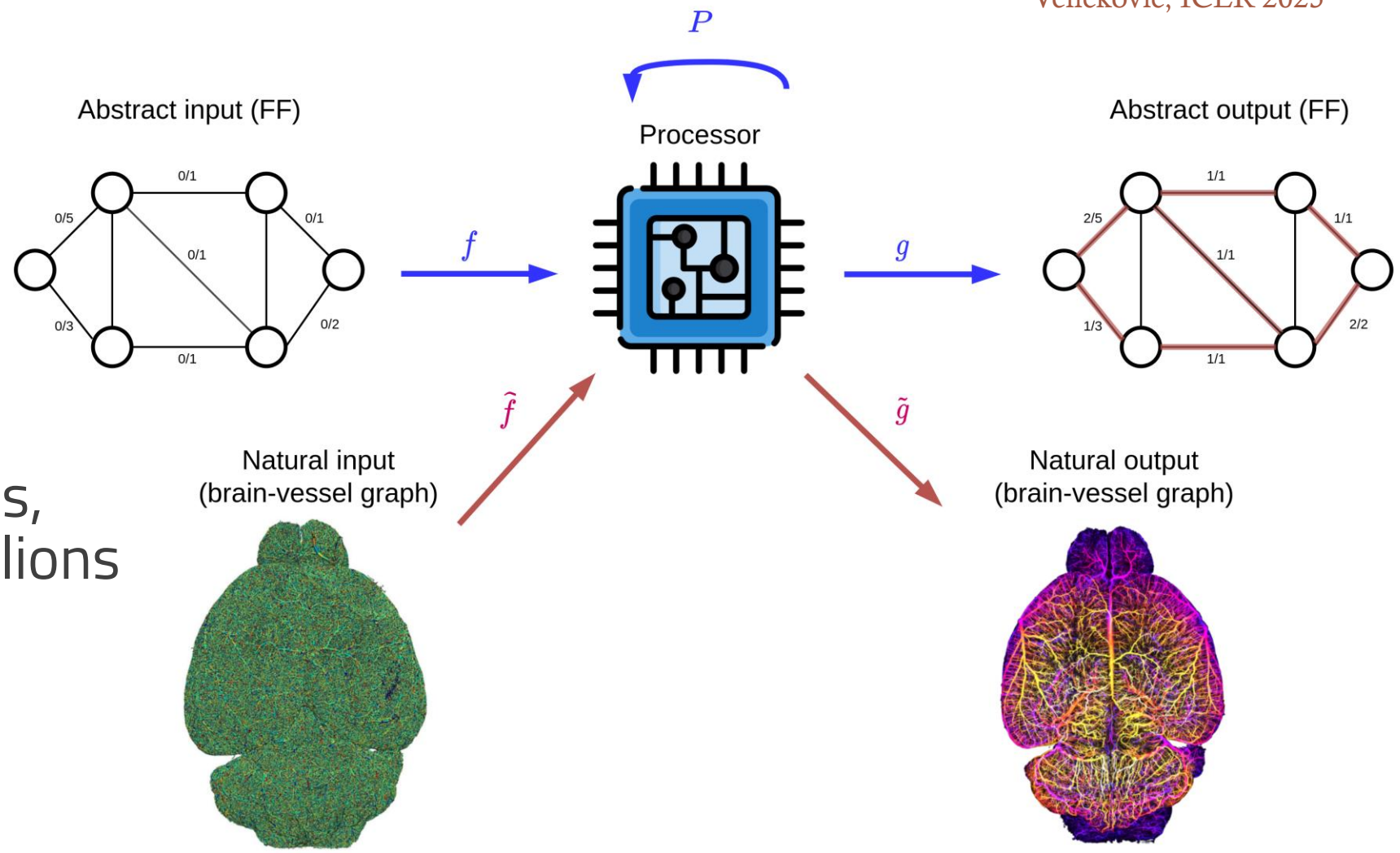
Example: Ford-Fulkerson, Max-Flow & Min-Cut



Learning min-cut is essential!
(Max-Flow/Min-Cut theorem)

Scaling up way out of distribution

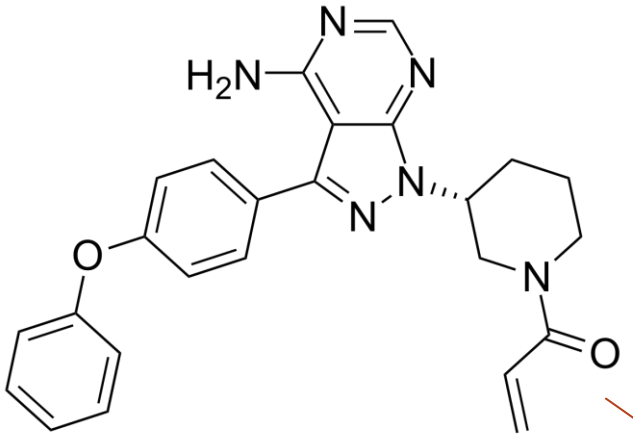
Numeroso, Bacciu, Velickovic, ICLR 2023

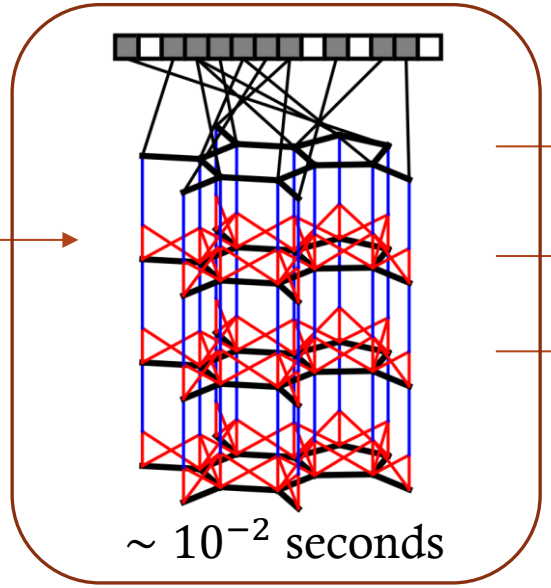


Train on 16 nodes, predict on 10 millions

Applications

Predicting Properties of Chemical Compounds





Toxicity

Solubility

Quantum mechanical properties

Simulation methods

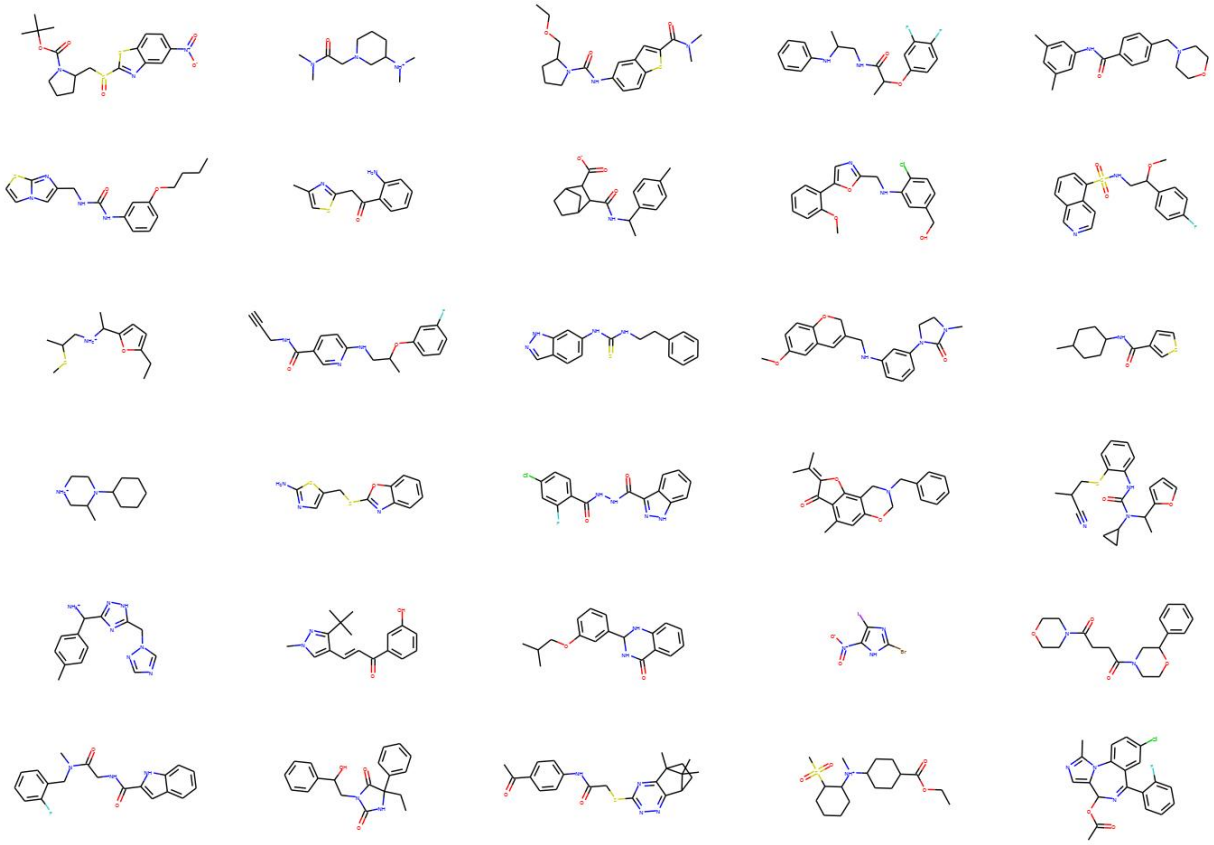
~ 10^{3} seconds

Micheli et al, JCICS 2001

Duvenaud, Maclaurin et al, NIPS 2015

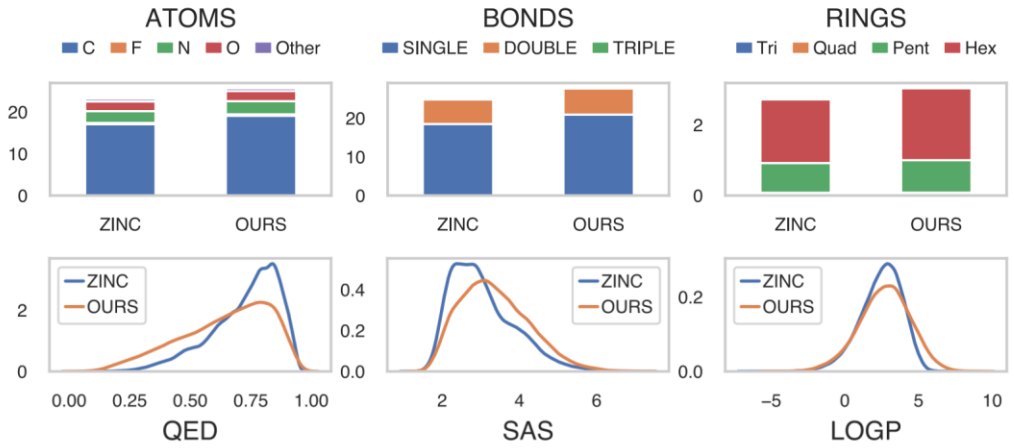
Gilmer et al, ICML 2017

Generating Molecules

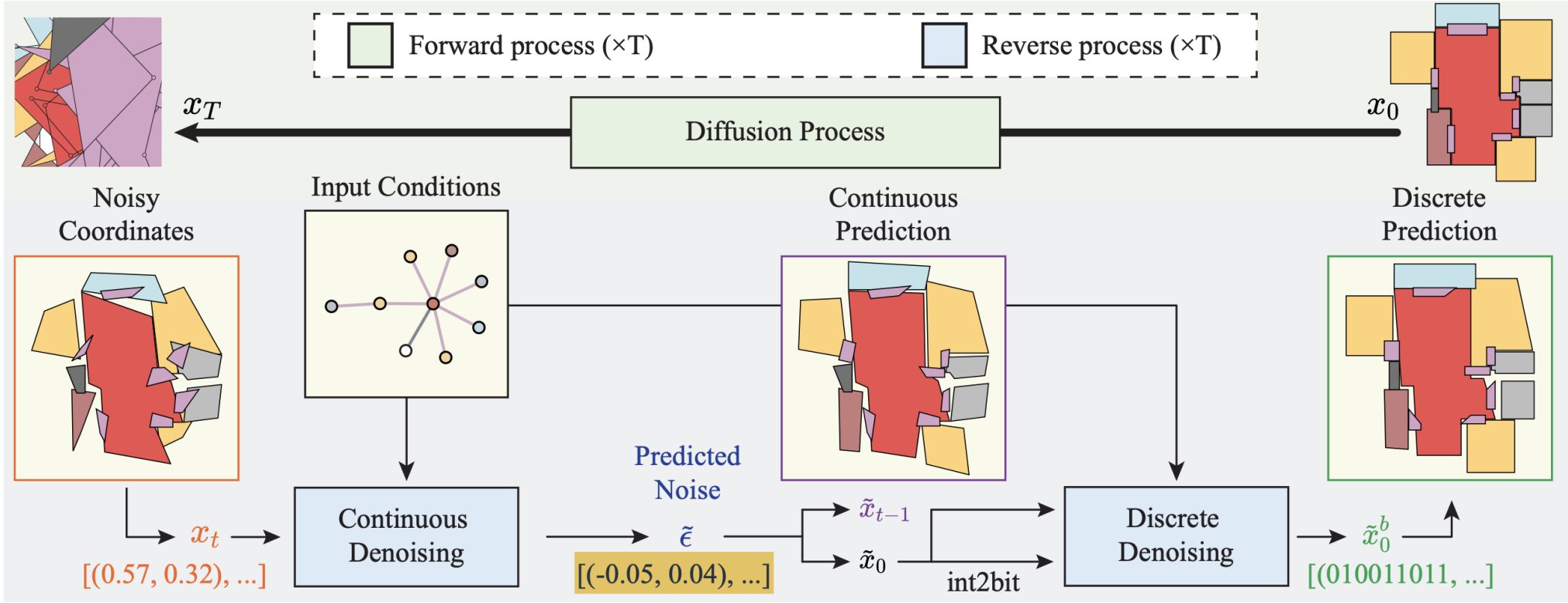


Podda, Bacciu, Micheli,
AISTATS 2020

Fragment-based deep molecule generation

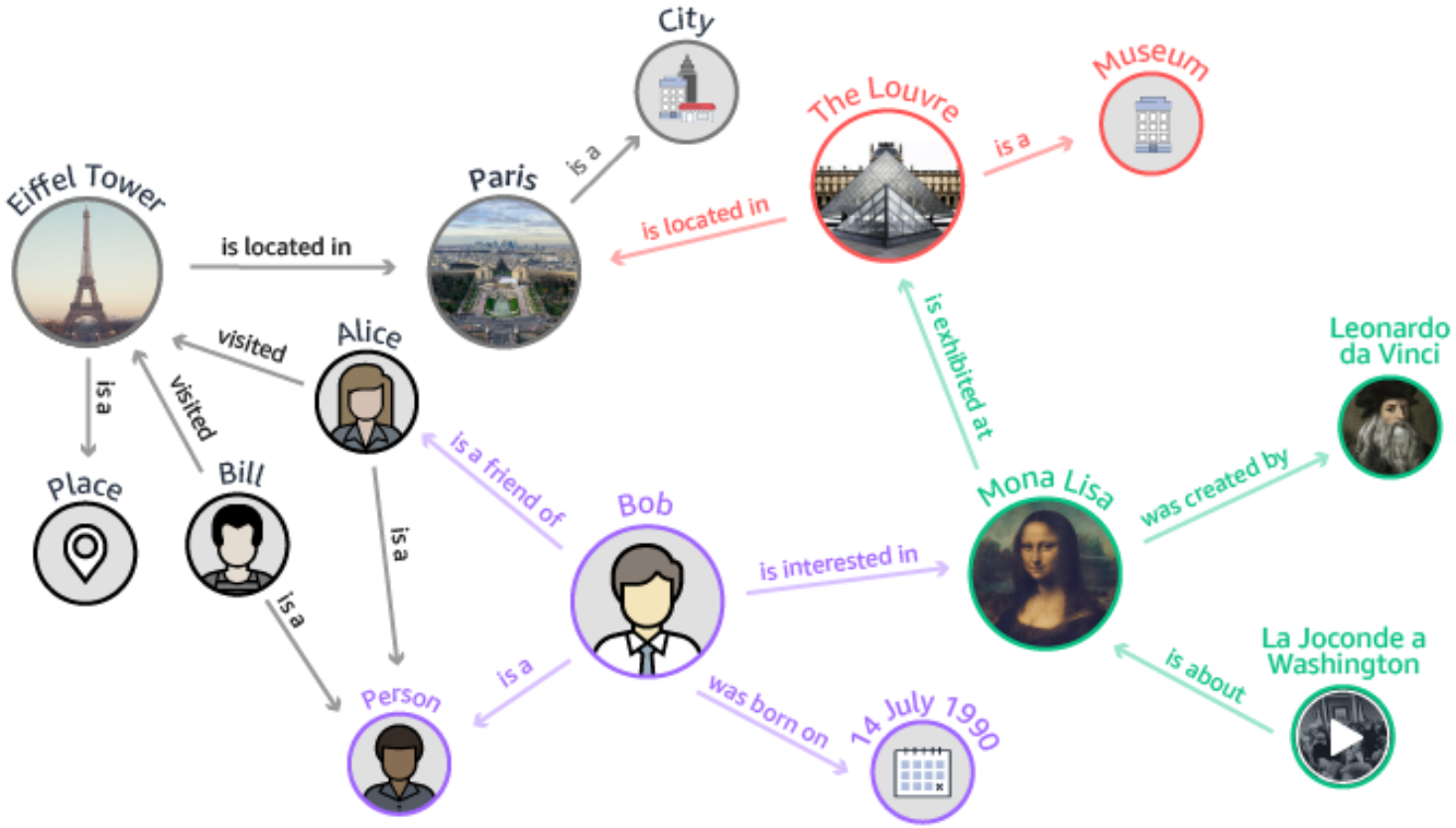


Floorplans Generation



Shabani et al, CVPR 2023

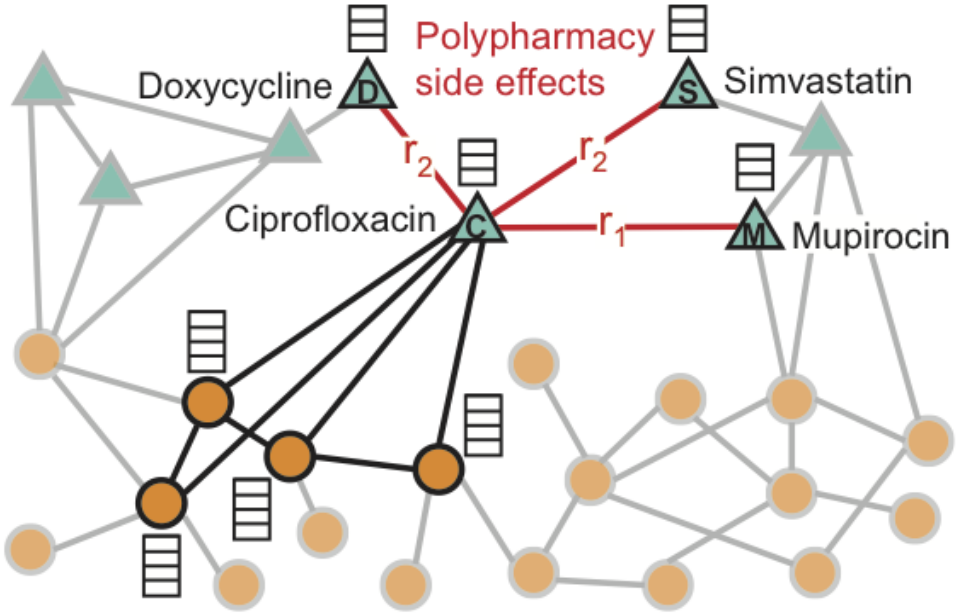
Knowledge graphs



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

Node/link embeddings are numerical encodings of entities and relationships

Side Effects of Drug Combinations



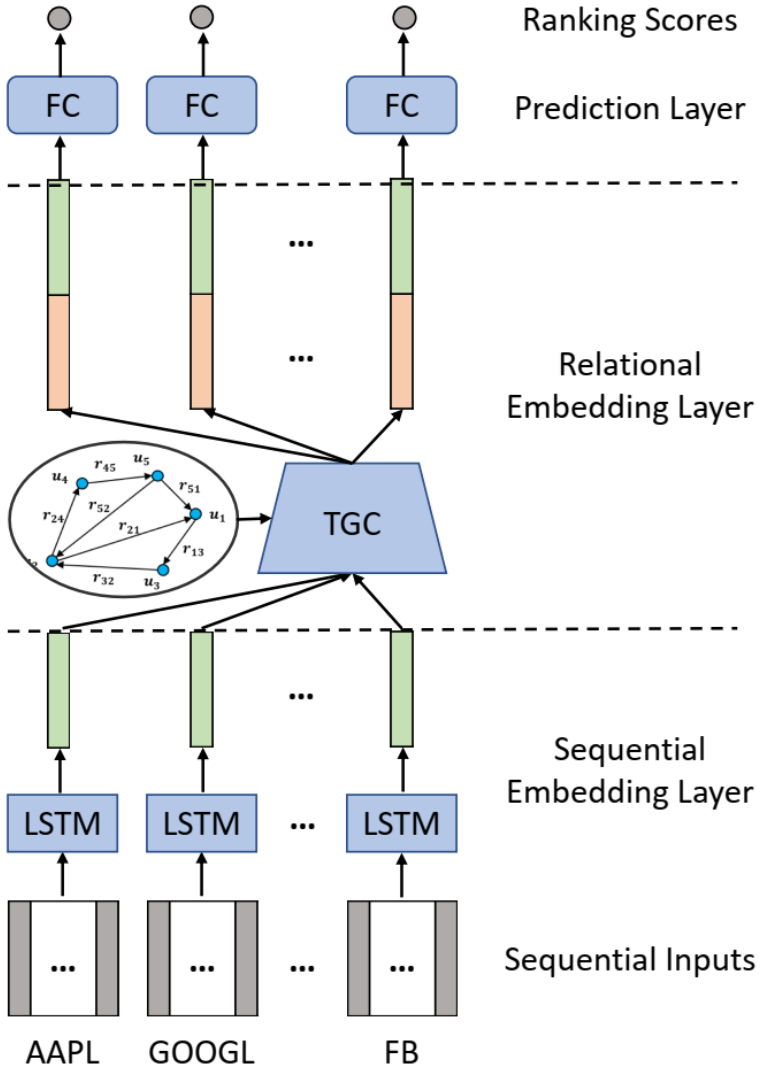
Analyzing a multimodal graph of interactions

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

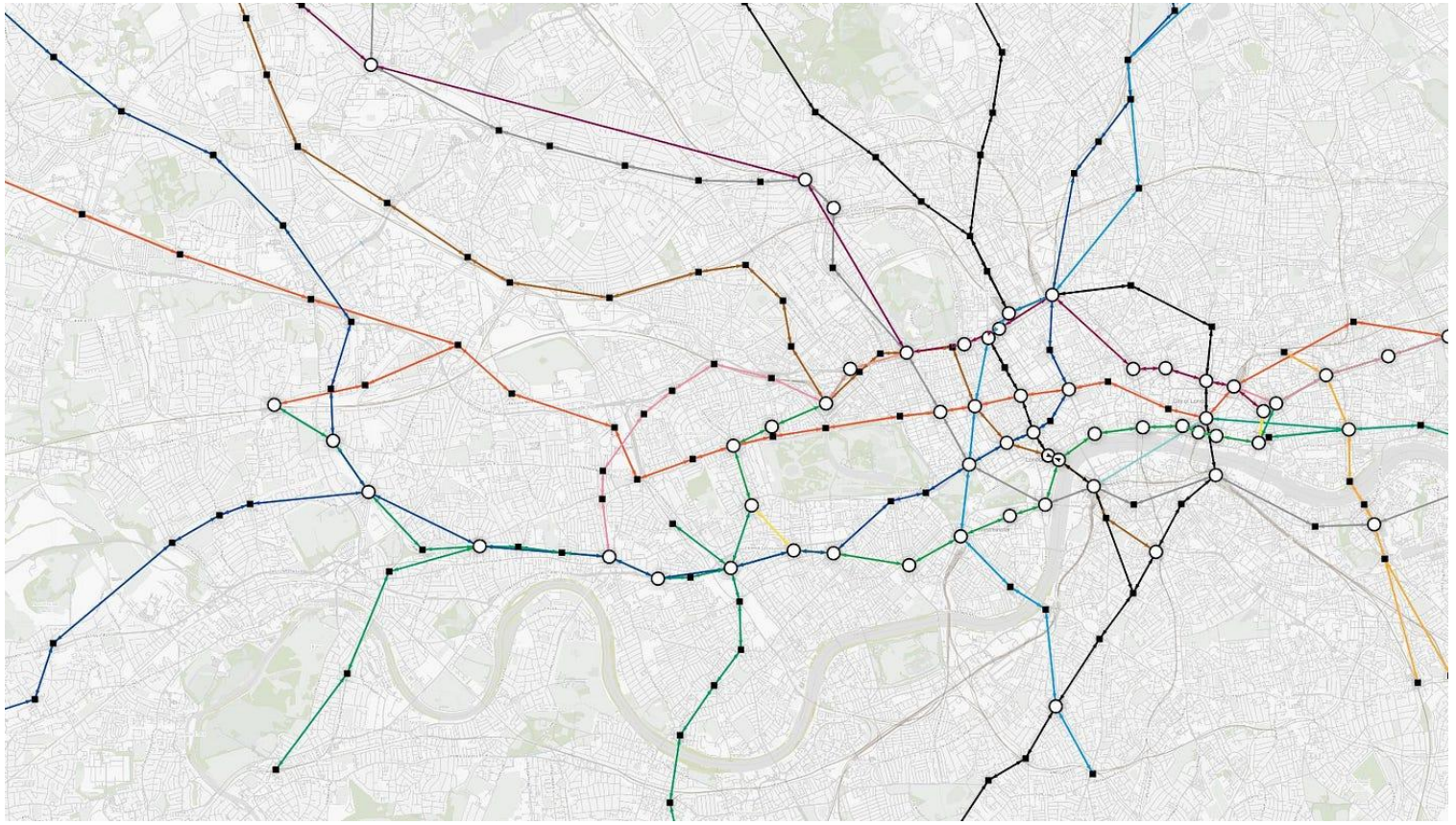
Drug
Protein
Node feature vector
r_1 Gastrointestinal bleed side effect
r_2 Bradycardia side effect
Drug-protein interaction
Protein-protein interaction

Zitnik, Agrawal, Leskovec, Bioinformatics 2018

Relational Stock Learning



Spatio-Temporal Transportation Networks



- Forecasting arrival times
- Identifying anomalies
- Route replanning

The content of this lecture is not part of the oral exam!

Wrap-up

Take home messages

- Generative learning for graphs is receiving growing attention, especially in connection with bio-chemical applications
  - Defining continuous processes over combinatorial data is non-trivial
  - Need careful thinking about differentiability
- Current research is heavily focusing on
  - Dynamic graphs and spatio-temporal networks processing
  - DGNs as dynamical systems and their physical interpretation
  - Learning and aligning with (graph) algorithms
  - Oversmoothing, oversquashing and problems of the sort
- …in other words, plenty of opportunities for thesis work!

Next Lecture

A super-compressed introduction to reinforcement learning

- RL Fundamentals
- Model based RL
- Model free RL
- Hints of deep reinforcement learning