FAST AND DEEP GRAPH NEURAL NETWORK[4]

Aliprandi Francesco

PROBLEM

WHY GRAPHS: graphs ar relevant data structures that provide an useful abstraction for many kind of real data (*molecules, social networks, transportation systems*…)

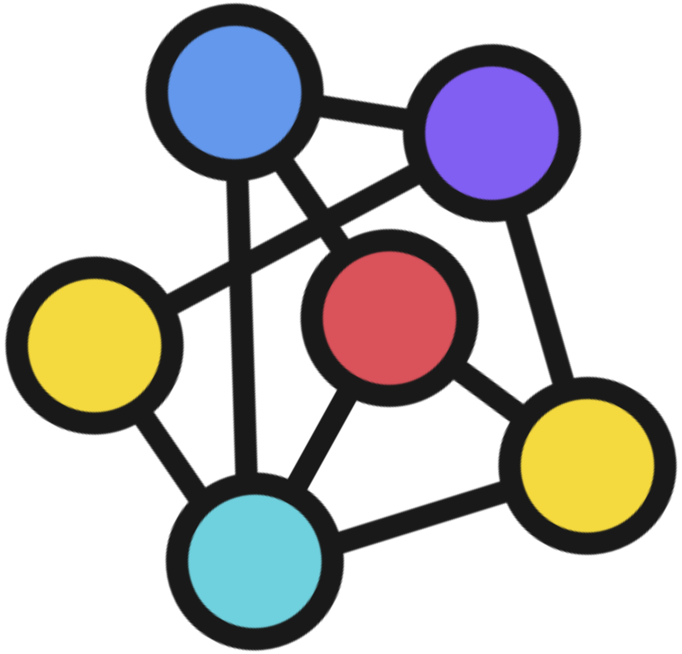
Graph Neural Networks (GNNs) have emerged as powerful tools for managing it [1]

from flat to structured domains

+

from shallow to deep architectures

→ richer representations but **high computational cost** [2]



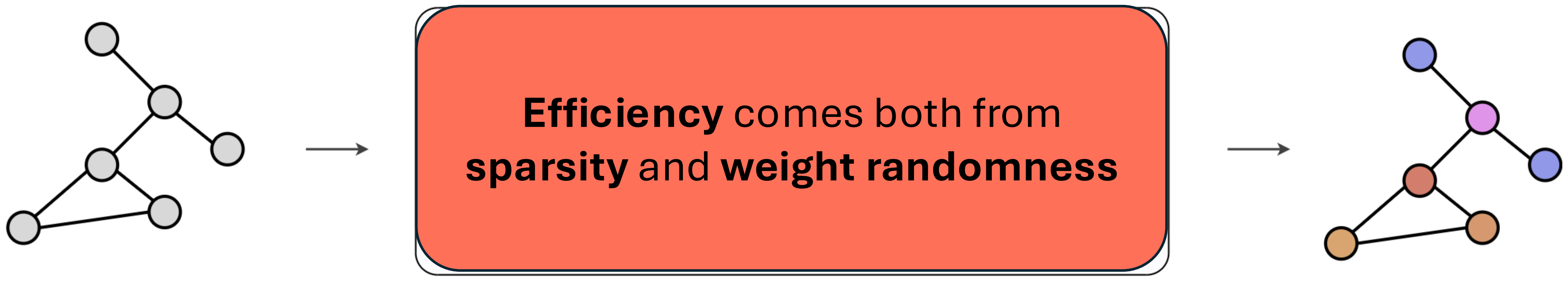
Solution: Reservoir Computing [3] + Recursive Processing of Graphs [4]

FAST AND DEEP GNN

It combines:

1. The capability of stable dynamic systems for the **graph embedding**
2. The potentiality of **deep organization** of the GNN architecture
3. The extreme efficiency of a randomized neural network

core idea is to exploit the fixed point of the recursive/dynamical system to embed the input graphs

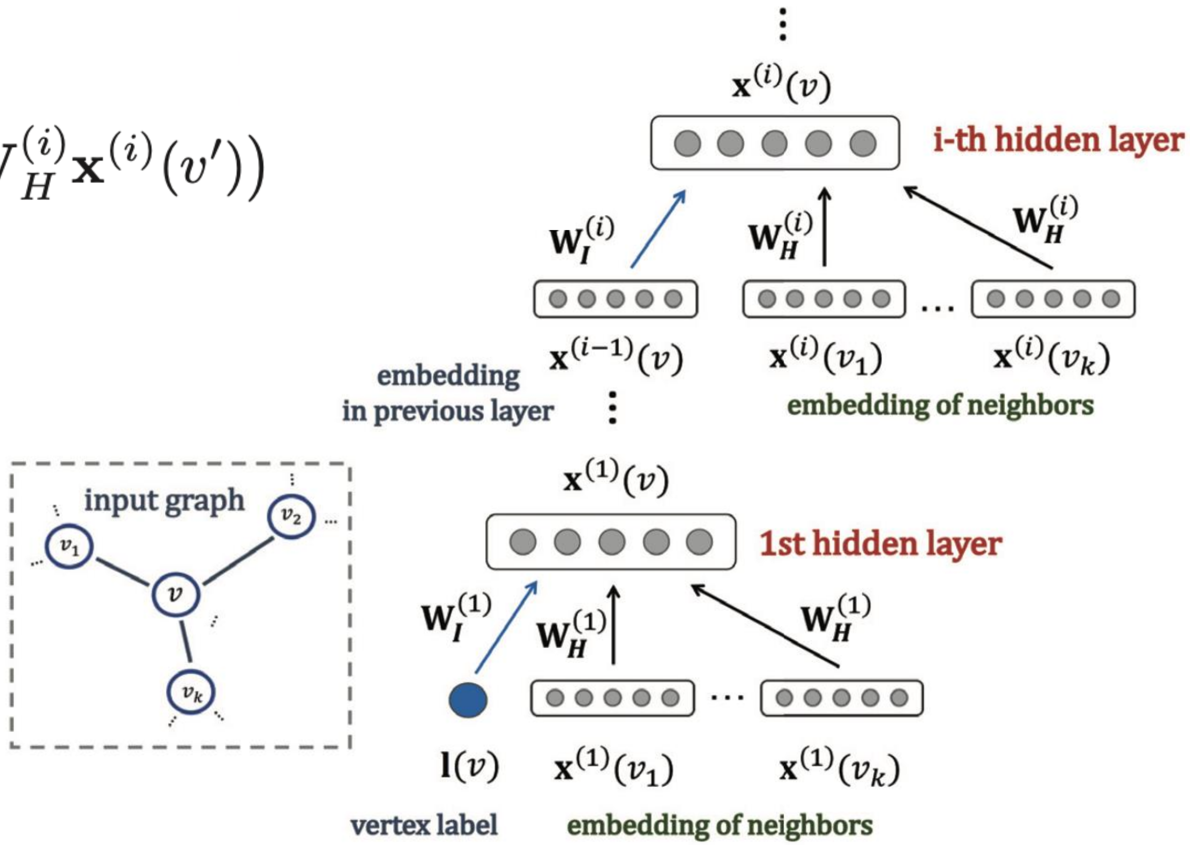


input graph — iterative state computation — fixed point embedding

PROPOSED METHOD

Problem: Given a **set graphs** *G = (V$_G$, E$_G$)* and its adjacency matrix $A_G \in R^{N_G \times N_G}$ with values in [0,1] we want to **classify** them

FDGNN [5]: $$\mathbf{x}^{(i)}(v) = \tanh\left(\mathbf{W}_I^{(i)}\mathbf{u}^{(i)}(v) + \sum_{v' \in \mathcal{N}(v)} \mathbf{W}_H^{(i)}\mathbf{x}^{(i)}(v')\right)$$



Where:

- i refers to the i-th layer
- $\boldsymbol{u}^{(i)}(v)$ is the current input:
  - $\boldsymbol{u}^{(1)}(v) = \boldsymbol{l}(v)$
  - $\boldsymbol{u}^{(i)}(v) = x^{(i-1)}(v)$

Initialization to 0 value for node embedding

PROPOSED METHOD

Using $\mathbf{U}^{(i)}$ and $\boldsymbol{X}^{(i)}$ as **column-wise** collection of $\boldsymbol{u}^{(i)}(v)$ and $\boldsymbol{x}^{(i)}(v)$:

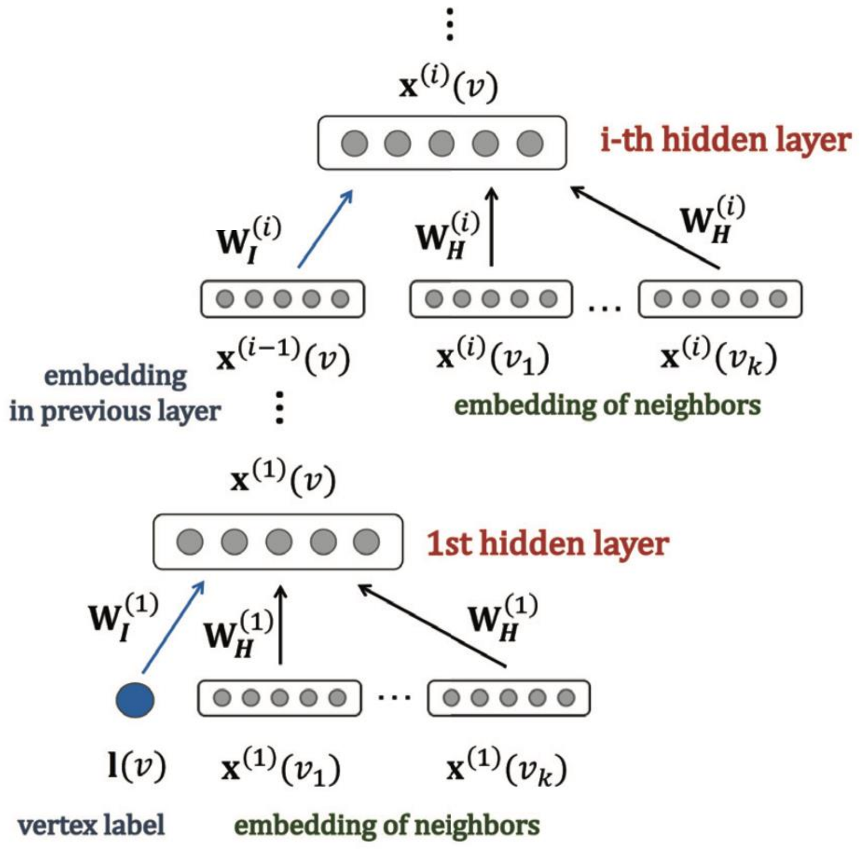
FDGNN:

$$\mathbf{X}^{(i)} = F^{(i)}(\mathbf{U}^{(i)}, \mathbf{X}^{(i)}) = \tanh(\mathbf{W}_I^{(i)}\mathbf{U}^{(i)} + \mathbf{W}_H^{(i)}\mathbf{X}^{(i)}\mathbf{A})$$

With $F^{(i)}: R^{U^{(i)} \times N} \times R^{H^{(i)} \times N} \to R^{H^{(i)} \times N}$

nota: in case of mutual dependencies among vertexes (cycles…) this equation might not admit a unique solution

To ensure **uniqueness of neural representation**, asymptotical stability is required. Both the input $\mathbf{U}^{(i)}$ and the adjacency matrix $\boldsymbol{A}$ plays a foundamental role



EMBEDDING PROCEDURE

Algorithm 1 Layer-wise State Convergence in GESN

Require: Graph $G = (V, E)$, vertex labels, number of layers L, threshold ε, max iterations ν

```
X^(0) ← vertex labels
for each layer i = 1 to L do
    X_0^(i) ← 0
    t ← 0
    repeat
        X_{t+1}^(i) ← F^(i)(U^(i), X_t^(i))
        t ← t + 1
    until ‖X_t^(i) − X_{t−1}^(i)‖ < ε or t ≥ ν
end for
return X^(L)
```

THE ONLY FREE PARAMETERS!

POOLING SUM FUNCTION FOR GRAPH-LEVEL TASK

Output computation: $\mathbf{y} = \mathbf{W}_Y \tanh\left(\mathbf{W}_\varphi \sum_{v \in V} \mathbf{x}^{(L)}(v)\right)$

Elements in W_ϕ are randomly initialized and rescaled to have unitary L2-norm

STABILITY CONDITION

Let F_t and X_t denote the function F and the state X at the $t-th$ iteration:

$$\boldsymbol{X}_t = F(\boldsymbol{U},\boldsymbol{X}_{t-1}) = F\big(\boldsymbol{U},F(\boldsymbol{U},\boldsymbol{X}_{t-2})\big) = \ \dots \ = F(\boldsymbol{U},F(\boldsymbol{U},F(\dots\big(F(\boldsymbol{U},\boldsymbol{X}_0)\big)\dots))).$$

Assuming that input and state space are **compact sets**

note: the focus of the analysis is on a generic layer i of the architecture so the index is dropped for the ease of notation

Graph Embedding Stability (GES)

Def: For every input $\boldsymbol{U}$ to the current layer, and for every X_0, Z_0 initial states for the neural embeddings in the current layer, it results that:

$$\|F_t(\mathbf{U},\mathbf{X}_0) - F_t(\mathbf{U},\mathbf{Z}_0)\| \to 0 \quad as \;\; t \to \infty.$$

ORIGINAL CONDITION

Sufficient condition for GES:

For every input $\boldsymbol{U}$ to the current layer, if $\| W_H \| \, k < 1$ then F has dynamics that satisfy the GES property.

(too restrictive in practical applications)

Necessary condition for GES:

Assume that a k-regular graph with null vertices labels is an admissible input for the system. If F has dynamics that satisfy the GES property, then $\rho(W_H)\, k < 1$.

This conditions rely on **local information** (k value) but a good bound will consider **global connectivity information.** For **undirected** graphs ($\boldsymbol{A}$ is symmetric) we consider:

$$\alpha^* = \rho(\boldsymbol{A}) = ||\boldsymbol{A}||, \qquad \text{because } \boldsymbol{A} \text{ symmetric}$$

The following conditions allows to develop a very effective dynamic, close to the edge of chaos [6]

With k the maximum among the sizes of the neighborhoods of the vertices

SUFFICIENT CONDITION

For every input $\boldsymbol{U}$ to the current layer, if $\| \boldsymbol{W}_H \| \, \alpha^* < 1$, where α^* is the input graph spectral norm, then F has dynamics that satisfy the GES property [9].

Proof:

Considering: $$\mathbf{X}_t = \begin{cases} \tanh\left((\mathbf{I} \otimes \mathbf{W}_I)\mathbf{U} + (\mathbf{A} \otimes \mathbf{W_H})\mathbf{X}_{t-1}\right) & t > 0 \\ \mathbf{X}_0 & t = 0 \end{cases}$$

Where: $$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}$$

Considering $\boldsymbol{I} \otimes \boldsymbol{W}_I$ as **input to reservoir weights** and $\boldsymbol{A} \otimes \boldsymbol{W}_H$ **as reservoir recurrent units**, given the **Lipschitz continuity** of F, consider the evolution of the state difference at **time** T when starting from x_0 and x_0':

$$\begin{aligned} \|F_T(\mathbf{u}, \mathbf{x}_0) - F_T(\mathbf{u}, \mathbf{x}_0')\| &= \left\|F(\mathbf{u}, \mathbf{x}_{T-1}) - F(\mathbf{u}, \mathbf{x}_{T-1}')\right\| \\ &\leq \|\mathbf{A} \otimes \mathbf{W}_H\| \left\|\mathbf{x}_{T-1} - \mathbf{x}_{T-1}'\right\| \\ &\dots \\ &\leq \|\mathbf{A} \otimes \mathbf{W}_H\|^T \|\mathbf{x}_0 - \mathbf{x}_0'\| \\ &= (\alpha^* \|\mathbf{W}_H\|)^T \|\mathbf{x}_0 - \mathbf{x}_0'\| . \end{aligned}$$

A perturbation on the initial state thus propagates through iterations as $(\alpha * \| W_H \|)^T$

NECESSARY CONDITION

If F has dynamics that satisfy the GES property under null input $\boldsymbol{u} = \mathbf{0}$, then $\rho(\boldsymbol{W}_H)\ \alpha^* < 1$, where α^* is the input graph spectral radius[9]

Proof:

Consider the linearised version of $$\mathbf{X}_t = \begin{cases} \tanh\left((\mathbf{I} \otimes \mathbf{W}_I)\mathbf{U} + (\mathbf{A} \otimes \mathbf{W_H})\mathbf{X}_{t-1}\right) & t > 0 \\ \mathbf{X}_0 & t = 0 \end{cases},$$

then around the zero state for null input, $\tilde{x} = (\boldsymbol{A} \otimes \boldsymbol{W}_H)\tilde{x}$. If the condition $\rho(\boldsymbol{A} \otimes \boldsymbol{W}_H) = \alpha * \rho(\boldsymbol{W}_H) < 1$ is violated, then the system is unstable around the zero state, and therefore the GES property is not satisfied

Remark 1: In both sufficient and necessary condition $\| \boldsymbol{A} \| = \rho(\boldsymbol{A}) = \alpha^*$, assuming that the input graph is undirected. Both can be extended to **directed graphs** as $\| \boldsymbol{W}_H \| < 1/\| \boldsymbol{A} \|$ and $\rho(\boldsymbol{W}_H) < 1/\rho(\boldsymbol{A})$**,** respectively

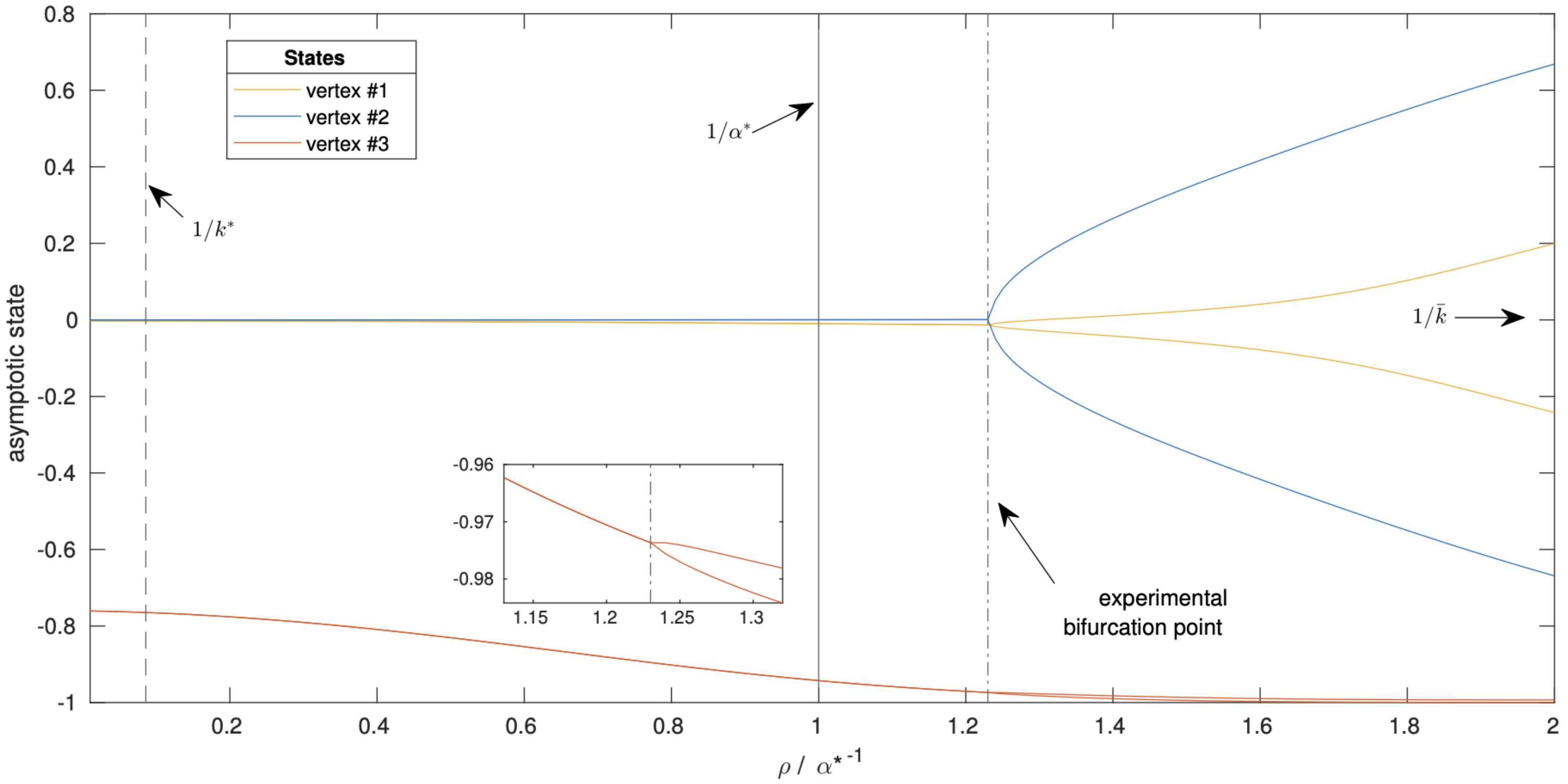
Remark 2: to deal with **weighted graph**, its enough to include the graph laplatian $(\boldsymbol{D} - \boldsymbol{A})$ in the equation, keeping valid GES conditions

With k the maximum among the sizes of the neighborhoods of the vertices

BIFURCATION DIAGRAM[9]

Setup:

- 1000 state transition function iterations
- single-unit reservoir
- different values of ρ
- mono-dimensional random vertex input feature



asymptotic state values for three vertices of the Cora graph, which represents the citation network of 2708 scientific publications [7]

$1/k^*$ **and** $1/avg(k)$ **provides good bounds for hyperparameter selection on desired spectral radius**

WEIGHT INITIALIZATION

Hidden-to-hidden weight initlization: randomly initialize from u.d. between [−1,1] and rescaled to have the desired spectral radius

Since each graph in the dataset has its own α^* and vertex degrees k^*, the standard practice [5], [8] consisits in using the average values across the graphs in the dataset to constrain **recurrent weights initialization**

Input to hidden weight initlization: randomly sample from u.d. between $\left[-\omega^{(i)}, \omega^{(i)}\right]$.

COMPUTATIONAL COST

FDGNN: $\mathbf{X}^{(i)} = F^{(i)}(\mathbf{U}^{(i)}, \mathbf{X}^{(i)}) = \tanh(\mathbf{W}_I^{(i)}\mathbf{U}^{(i)} + \mathbf{W}_H^{(i)}\mathbf{X}^{(i)}\mathbf{A})$

Assuming no more than C connections among neurons in hidden layers (sparsity), the cost of computing $\boldsymbol{X}^{(i)}$ for each layer is: $\mathcal{O}((C+k)\,H\,N)$

The entire process of graph embedding cost is: $\mathcal{O}(L\,\nu\,(C+k)\,H\,N)$

With ν the max number of iterations

cost of the encoding is the same for both training and test

Readout trained by direct method for efficiency reasons.

EXPERIMENTS

FDGNN was evaluated and compared with s.o.a. model on **9 public benchmark datasets** for **graph classification**

Datasets comes from **cheminformatics** (*dataset of proteins that are classified as enzymes or non-enzymes*) and **Social Networks** (*e.g. movie collaboration datasets containing actor/actress, target is the movie genre*)

Experimental setting:

- all the hidden layers of the architecture in the graph embedding component shared the same values of the hyper-parameters
- H fixed to 50 and 500 according to the datset
- $C = 1$ each hidden neuron has one feedforward and one recurrent connection
- k is selected as the average on the graphs in the dataset (here we use k for condition stability, not α^*

RESULTS

Test accuracy averaged over the 10 folds of the cross-validation

	MUTAG	PTC	COX2	PROTEINS	NCI1	IMDB-b	IMDB-m	REDDIT	COLLAB
FDGNN	**88.51**$_{\pm 3.77}$	**63.43**$_{\pm 5.35}$	**83.39**$_{\pm 2.88}$	**76.77**$_{\pm 2.86}$	77.81$_{\pm 1.62}$	**72.36**$_{\pm 3.63}$	**50.03**$_{\pm 1.25}$	**89.48**$_{\pm 1.00}$	74.44$_{\pm 2.02}$
FDGNN$_{(L=1)}$	87.38$_{\pm 6.55}$	**63.43**$_{\pm 5.35}$	82.41$_{\pm 2.67}$	**76.77**$_{\pm 2.86}$	77.11$_{\pm 1.52}$	71.79$_{\pm 3.37}$	49.34$_{\pm 1.70}$	87.74$_{\pm 1.61}$	73.82$_{\pm 2.32}$
GNN (Uwents et al. 2011)	80.49$_{\pm 0.81}$	-	-	-	-	-	-	-	-
RelNN (Uwents et al. 2011)	87.77$_{\pm 2.48}$	-	-	-	-	-	-	-	-
DGCNN (Zhang et al. 2018)	85.83$_{\pm 1.66}$	58.59$_{\pm 2.47}$	-	75.54$_{\pm 0.94}$	74.44$_{\pm 0.47}$	70.03$_{\pm 0.86}$	47.83$_{\pm 0.85}$	-	73.76$_{\pm 0.49}$
PGC-DGCNN (Tran, Navarin, and Sperduti 2018)	87.22$_{\pm 1.43}$	61.06$_{\pm 1.83}$	-	76.45$_{\pm 1.02}$	76.13$_{\pm 0.73}$	71.62$_{\pm 1.22}$	47.25$_{\pm 1.44}$	-	**75.00**$_{\pm 0.58}$
DCNN (Tran, Navarin, and Sperduti 2018)	-	-	-	61.29$_{\pm 1.60}$	56.61$_{\pm 1.04}$	-	-	-	-
PSCN (Tran, Navarin, and Sperduti 2018)	-	-	-	75.00$_{\pm 2.51}$	76.34$_{\pm 1.68}$	71.00$_{\pm 2.29}$	45.23$_{\pm 2.84}$	-	72.60$_{\pm 2.15}$
GK (Zhang et al. 2018)	81.39$_{\pm 1.74}$	55.65$_{\pm 0.46}$	-	71.39$_{\pm 0.31}$	62.49$_{\pm 0.27}$	65.87$_{\pm 0.98}$	43.89$_{\pm 0.38}$	77.34$_{\pm 0.18}$	72.84$_{\pm 0.56}$
DGK (Yanardag and Vishwanathan 2015)	82.66$_{\pm 1.45}$	57.32$_{\pm 1.13}$	-	71.68$_{\pm 0.50}$	62.48$_{\pm 0.25}$	66.96$_{\pm 0.56}$	44.55$_{\pm 0.52}$	78.04$_{\pm 0.39}$	73.09$_{\pm 0.25}$
RW (Zhang et al. 2018)	79.17$_{\pm 2.07}$	55.91$_{\pm 0.32}$	-	59.57$_{\pm 0.09}$	-	-	-	-	-
PK (Zhang et al. 2018)	76.00$_{\pm 2.69}$	59.50$_{\pm 2.44}$	81.00$_{\pm 0.20}$	73.68$_{\pm 0.68}$	82.54$_{\pm 0.47}$	-	-	-	-
WL (Zhang et al. 2018)	84.11$_{\pm 1.91}$	57.97$_{\pm 2.49}$	83.20$_{\pm 0.20}$	74.68$_{\pm 0.49}$	**84.46**$_{\pm 0.45}$	-	-	-	-
KCNN (Nikolentzos et al. 2018)	-	62.94$_{\pm 1.69}$	-	75.76$_{\pm 0.28}$	77.21$_{\pm 0.22}$	71.45$_{\pm 0.15}$	47.46$_{\pm 0.21}$	81.85$_{\pm 0.12}$	74.93$_{\pm 0.14}$
CGMM (Bacciu, Errica, and Micheli 2018)	85.30	-	-	-	-	-	-	-	-

Typically from 3 to 5 layers are enough effective

RESULTS

Execution time of FDGNN on single core, without GPU

Task	Training	Test
MUTAG	$0.56''_{\pm 0.33}$	$0.06''_{\pm 0.04}$
PTC	$0.16''_{\pm 0.03}$	$0.02''_{\pm 0.00}$
COX2	$1.36''_{\pm 0.42}$	$0.15''_{\pm 0.05}$
PROTEINS	$2.16''_{\pm 0.47}$	$0.24''_{\pm 0.04}$
NCI1	$2.00'_{\pm 0.45}$	$13.36''_{\pm 3.02}$
IMDB-b	$7.46''_{\pm 3.14}$	$0.83''_{\pm 0.35}$
IMDB-m	$8.68''_{\pm 1.73}$	$0.98''_{\pm 0.22}$
REDDIT	$2.47'_{\pm 0.01}$	$16.49''_{\pm 0.28}$
COLLAB	$22.86'_{\pm 4.70}$	$2.54'_{\pm 0.52}$

FDGNN execution time compared with s.o.a GNN models

FDGNN	GNN	GIN	WL
$\mathbf{0.56''}_{\pm 0.33}$	$202.28''_{\pm 166.87}$	$499.24''_{\pm 2.25}$	$1.16''_{\pm 0.03}$
	361x	872x	2x

RESULTS UNDER NEW CONDITIONS

TABLE II
HOLD-OUT ACCURACY PEAKS FOR DIFFERENT RESERVOIR RADII

Task	$1/k_*$	$1/\alpha^*$	$1/\bar{k}$
NCI1	$75.6_{\pm 0.7}$	$77.2_{\pm 0.7}$	$77.8_{\pm 0.6}$
IMDB-Binary	$71.3_{\pm 0.7}$	$69.3_{\pm 1.0}$	$68.7_{\pm 1.2}$
Reddit-Binary	$78.0_{\pm 0.6}$	$89.0_{\pm 0.5}$	$82.6_{\pm 1.2}$
Reddit-Multi-5K	$53.4_{\pm 0.5}$	$57.2_{\pm 0.7}$	$52.7_{\pm 0.9}$
Reddit-Multi-12K	$33.4_{\pm 1.5}$	$42.9_{\pm 0.8}$	$35.3_{\pm 1.2}$

- distribution of **NCI1** is tightly concentrated around the mean
- **Reddit datasets** exhibit a long tail in α∗ distribution, with their means skewed towards larger values: instability produced by larger spectral redius graphs is compensated by the high number of small spectral radius that are moving toward EOS.

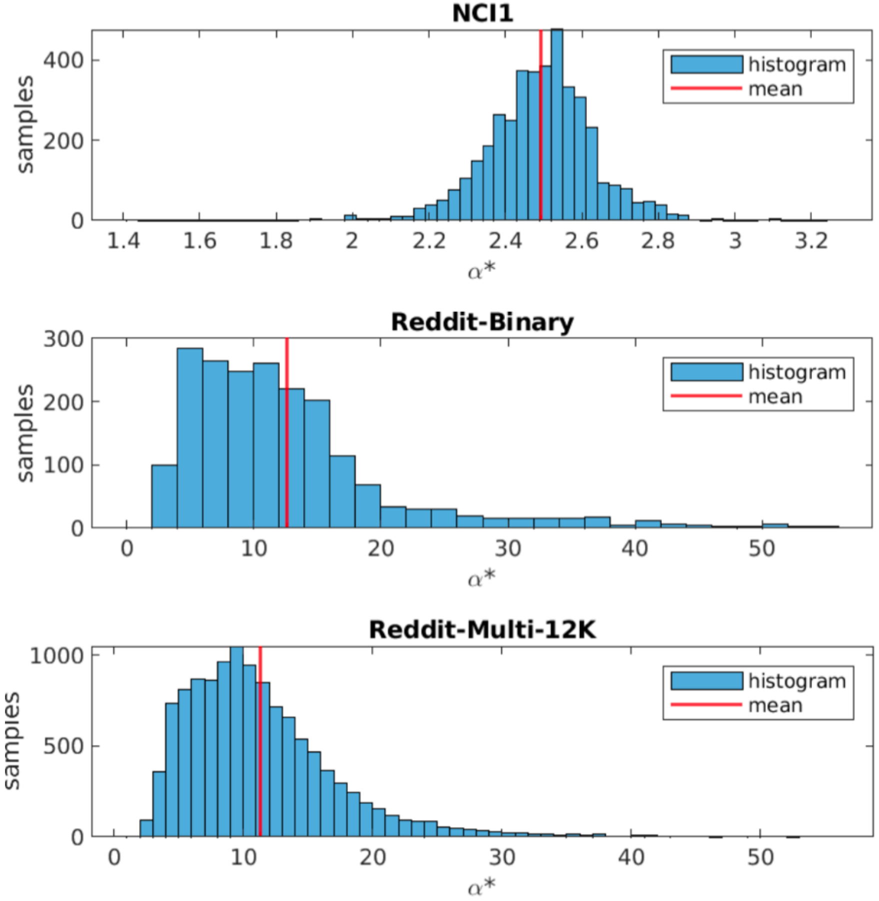
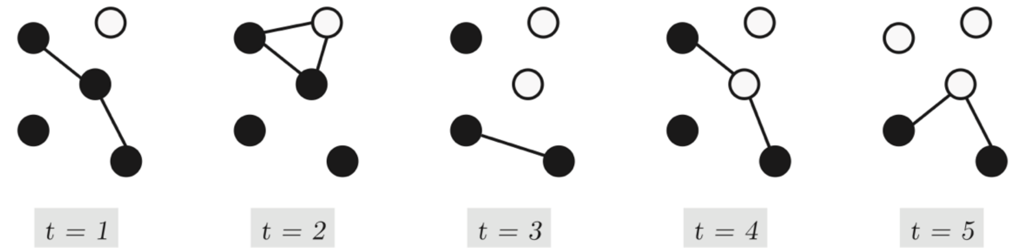


Fig. 4. Distribution of α^* on three datasets.

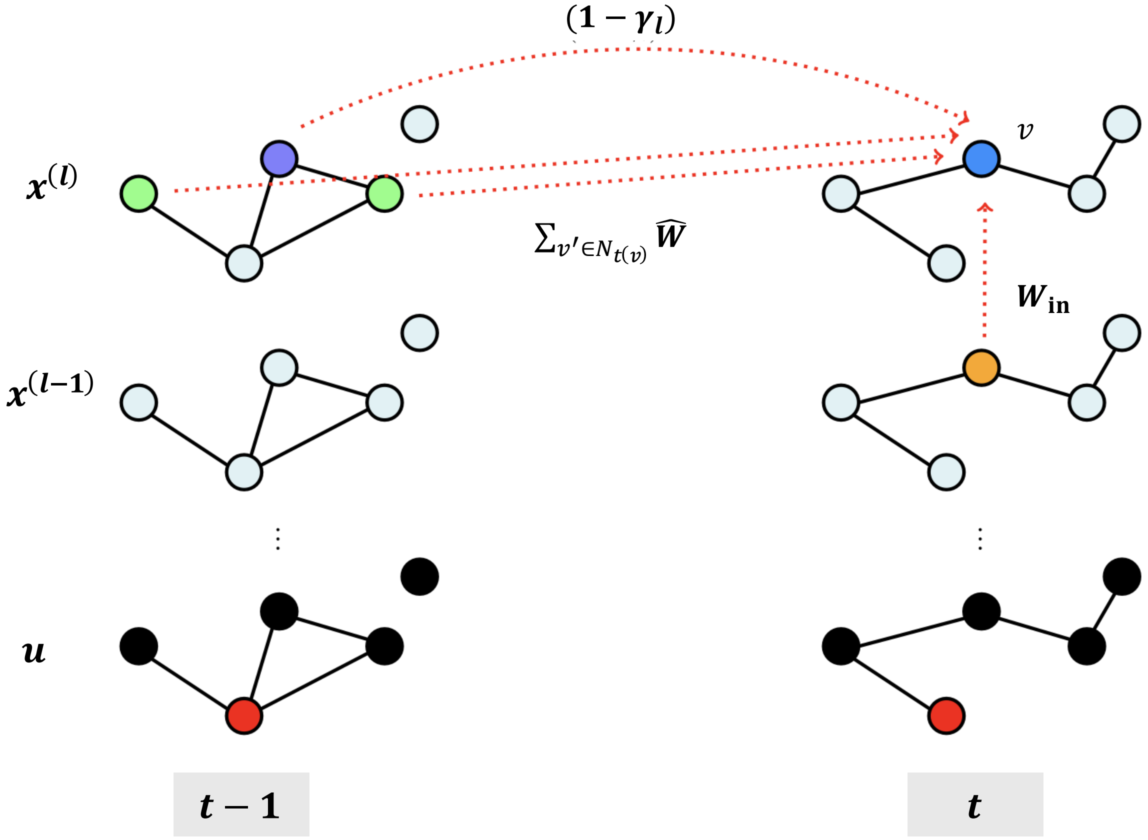
A FARTHER STEP... DynGESN



A **dynamic graph** G as a pair (V, E) with $E = \{(v_1; v_2; t) | v_1; v_2 \in V; t \in 1 \dots T\}$ the set of edges $v_1 \rightarrow v_2$ between a pair of vertices at a time-step t. The graph is **static** if graph topology is preserved across time, **dynamic** otherwise. A discrete-time dynamic graph can also be viewed as a **sequence of static graphs**.

DynGESN [10] **vertex-wise** definition:

$$\mathbf{x}_t^{(l)}(v) = \gamma_l \tanh\left(\mathbf{W}_{\text{in}}^{(l)}\mathbf{x}_t^{(\ell-1)}(v) + \sum_{v' \in N_t(v)} \hat{\mathbf{W}}^{(l)}\mathbf{x}_{t-1}^{(l)}(v')\right) + (1-\gamma_l)\mathbf{x}_{t-1}^{(l)}(v)$$



Graph embedding via **pooling operation** using final final state for each layer:

$$\mathbf{X}_{\mathcal{G}} = \begin{bmatrix} \sum_{v \in \mathcal{V}} \mathbf{x}_T^{(1)}(v) \\ \vdots \\ \sum_{v \in \mathcal{V}} \mathbf{x}_T^{(L)}(v) \end{bmatrix} \in \mathcal{X}^L \subset \mathbb{R}^{HL}$$

CONCLUSION

Novelties:

- A **fast** and deep GNN model: combining RC with deep GNN architectures
- A **better bound on GES property**

Weakness:

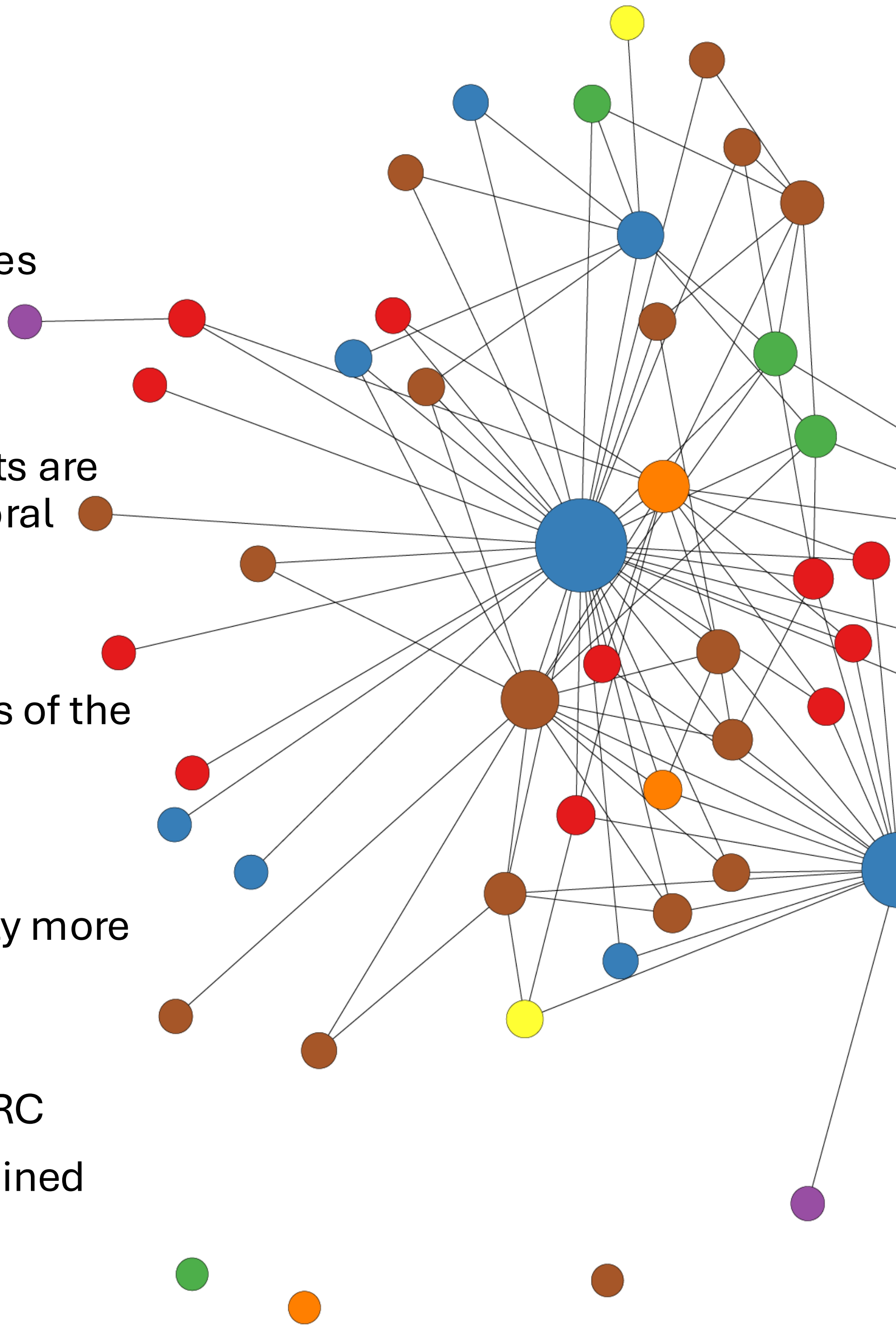
- **Limited Expressiveness** Compared to e2e GNN: since reservoir weights are fixed, Graph ESNs can underperform on tasks requiring adaptive temporal modeling or long-term dependencies

Strengths:

- Better bounds allows to better target the selection of the spectral radius of the reservoir towards the stability limit
- **fast** and **stable** training under GES property
- through the **deep architecture** the model is able to build a progressively more effective representation of the input graphs

Remarks:

- Model selection is the most **crucial** and **time consuming** operation in RC
- This bound bound can be applied in the **initialization of end-to-end** trained message passing models



BIBLIOGRAPHY

[1] D. Bacciu, F. Errica, A. Micheli, and M. Podda, “A gentle introduction to deep learning for graphs,” Neural Networks, vol. 129, pp. 203–221, 2020

[2]R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, “Green AI,” Communications of the ACM, vol. 63, no. 12, pp. 54–63, nov 2020

[3] M. Lukosevicius and H. Jaeger, “Reservoir computing approaches to recurrent neural network training,” Computer Science Review, vol. 3, no. 3, pp. 127–149, 2009

[4] C. Gallicchio and A. Micheli, “Graph echo state networks,” in The 2010 International Joint Conference on Neural Networks (IJCNN). IEEE, July 2010, pp. 3967–3974

[5] C. Gallicchio and A. Micheli, “Fast and deep graph neural networks,” in The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20), November 2020

[6] J. Boedecker, O. Obst, J. T. Lizier, N. M. Mayer, and M. Asada, “Information processing in echo state networks at the edge of chaos,” Theory in Biosciences, vol. 131, no. 3, pp. 205–213, 2012

[7] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, “Automating the Construction of Internet Portals with Machine Learning,” Information Retrieval, vol. 3, no. 2, pp. 127–163, 2000

[8] C. Gallicchio and A. Micheli, “Ring reservoir neural networks for graphs,” in 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020

[9] D. Tortorella, C. Gallicchio, A. Micheli, Spectral bounds for graph echo state net- work stability, in: The 2022 International Joint Conference on Neural Networks, 2022.

[10] A. Micheli, D. Tortorella, Discrete-time dynamic graph echo state networks, Neurocomputing 496 (2022) 85–95.

Note: papers in bold refers to the ones used as baseline for the seminar