Improving the robustness of Graph Neural Networks with coupled dynamical systems

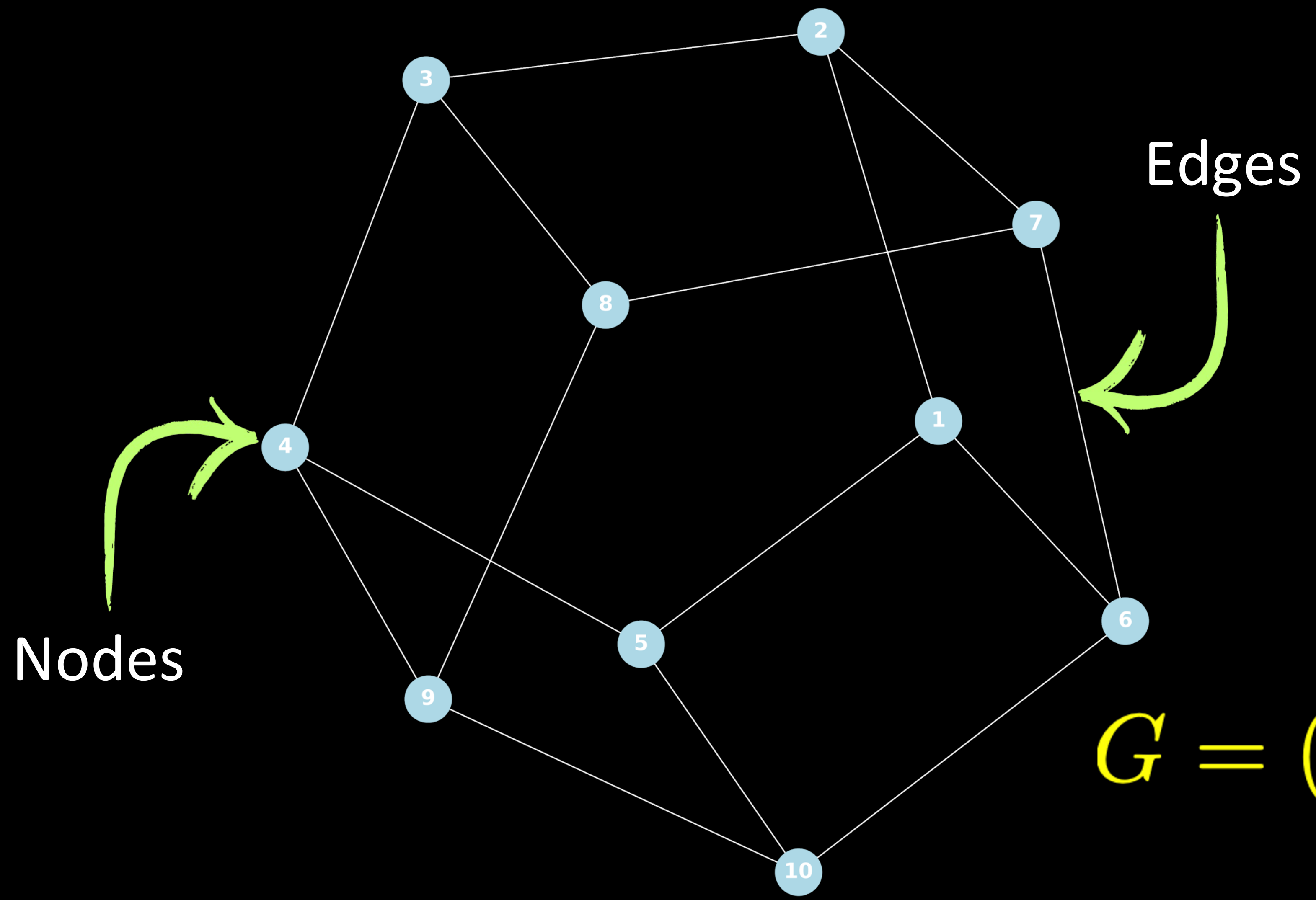
Davide Murari (NTNU)
davide.murari@ntnu.no

Structured Machine Learning and Time–Stepping for Dynamical Systems
Banff, 20/02/2024

In collaboration with Moshe Eliasof, Ferdia Sherry and Carola Schönlieb

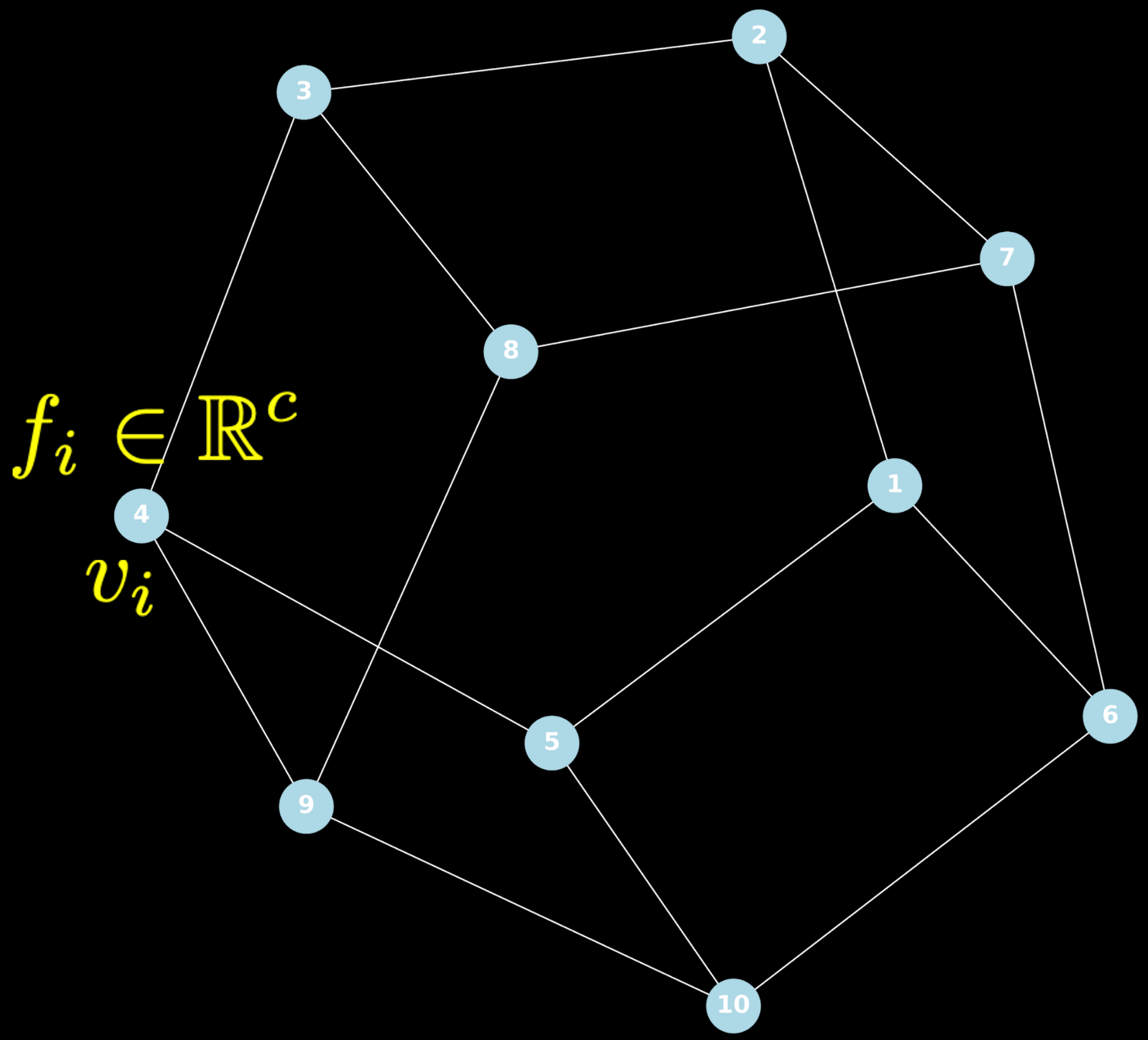
Outline

1. Brief introduction to graphs and Graph Neural Networks (GNNs)
2. The problem of adversarial robustness
3. Analysis of the GNN we propose
4. Numerical experiments



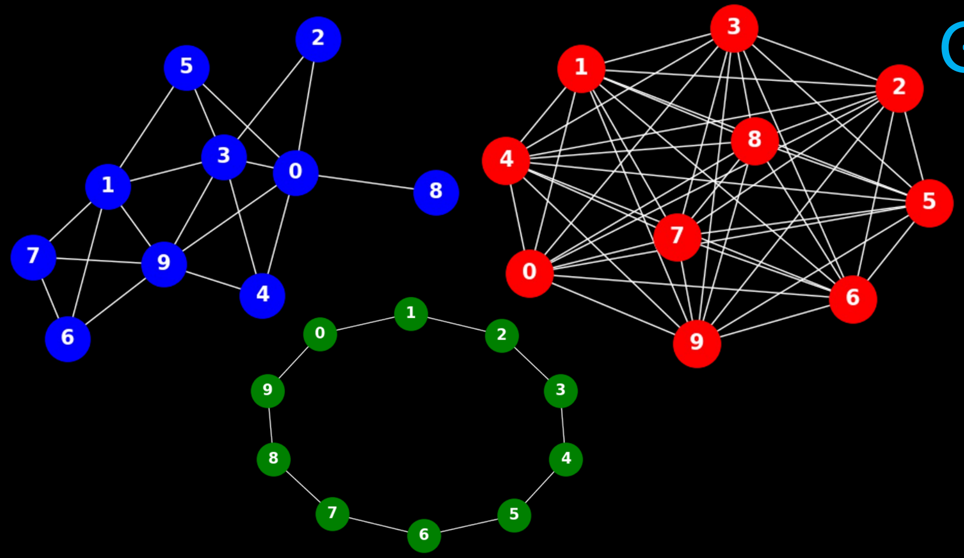
$$G = (V, E)$$

$$A \in \{0,1\}^{n \times n}, \;\; A^T = A$$



$$F \in \mathbb{R}^{n \times c}, e_i^T F = f_i$$

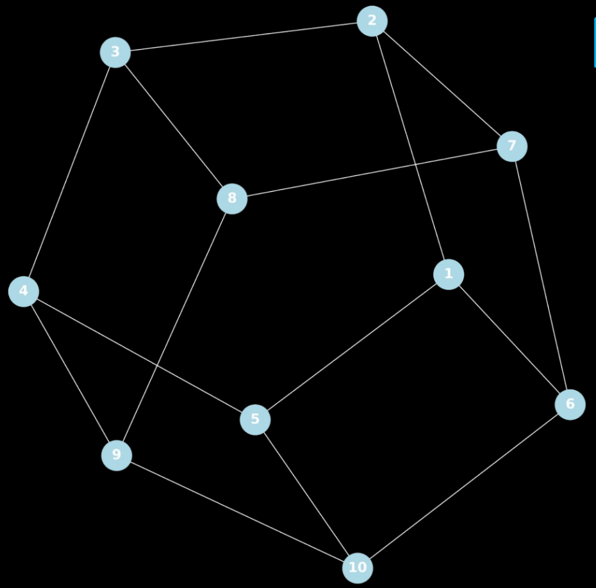
Classical tasks solved with GNNs



Graph classification

GNN

It is a protein
It is not a protein
It is a protein



Node classification

GNN

Red
Blue
Blue
...
Green

Usual structure of GNNs

$$F^{(0)} = F$$

$$F^{(l+1)} = T_l\left(F^{(l)}, A\right), l = 0, \ldots L - 1$$

$$R = \mathrm{MLP}\left(F^{(L)}\right) =: \mathrm{GNN}(F, A)$$

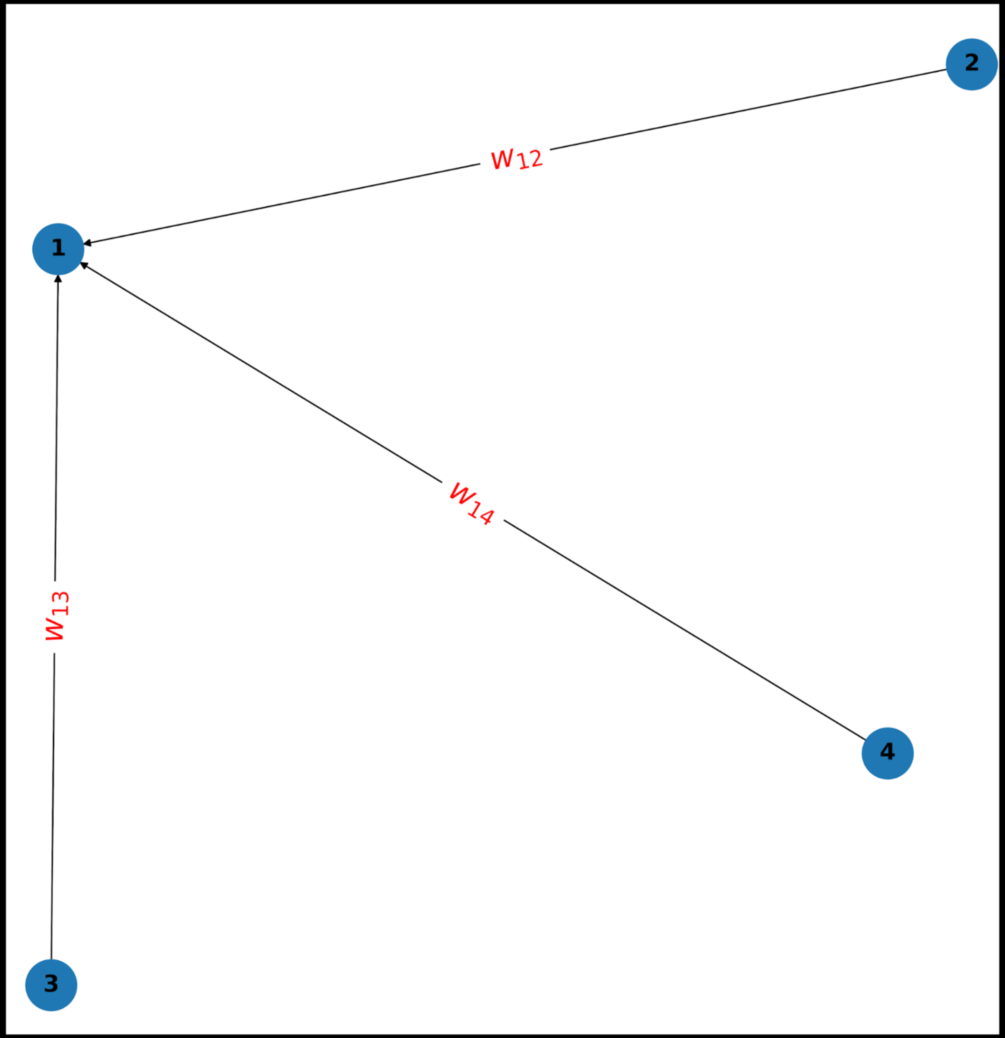
Invariant

$$\mathrm{GNN}(F, A) = \mathrm{GNN}(PF, PAP^T)$$

$$P\,\mathrm{GNN}(F, A) = \mathrm{GNN}(PF, PAP^T)$$

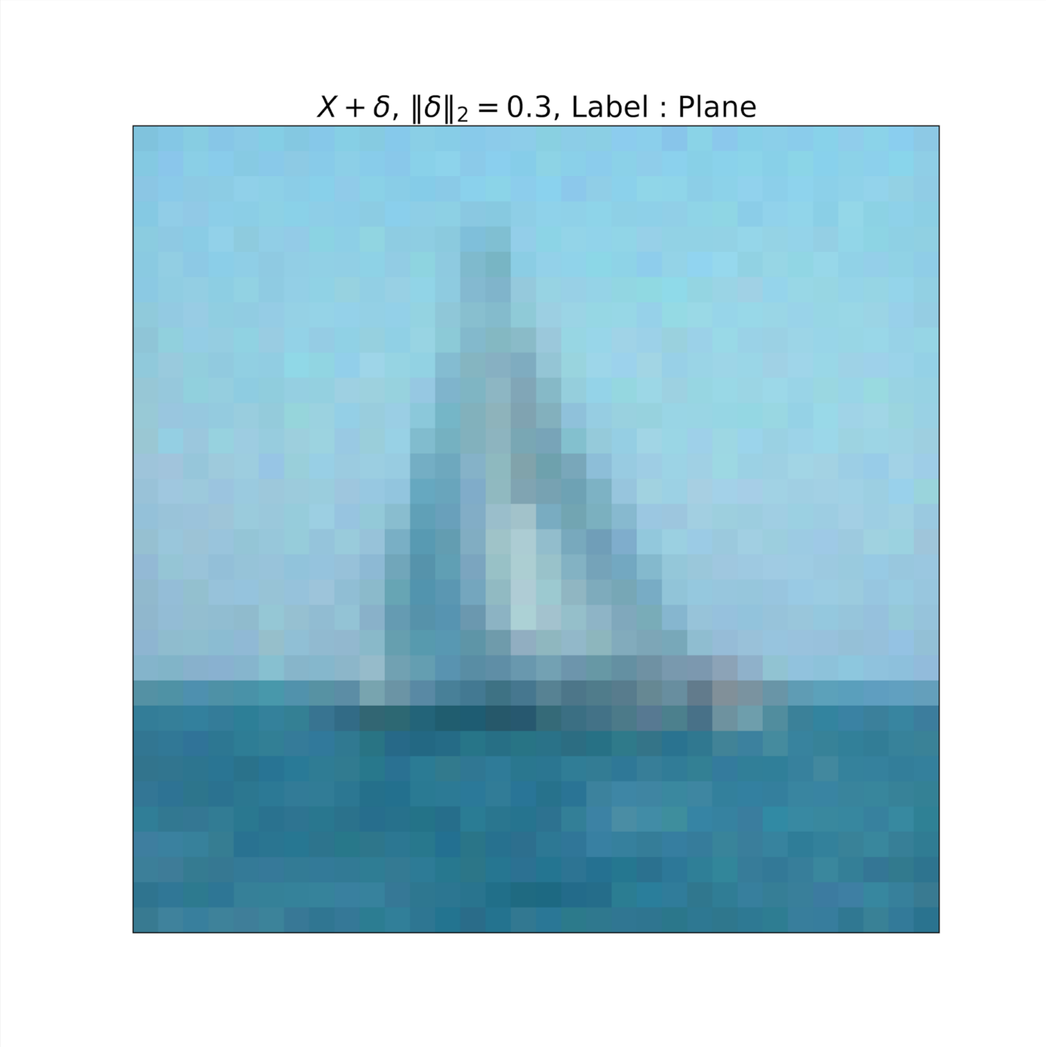
Equivariant

Simple example of a GNN update



$$f_1 = \phi\left(f_1, \sum_{j=2}^{4} w_{1j}\psi(f_j)\right)$$

Adversarial Attacks for images

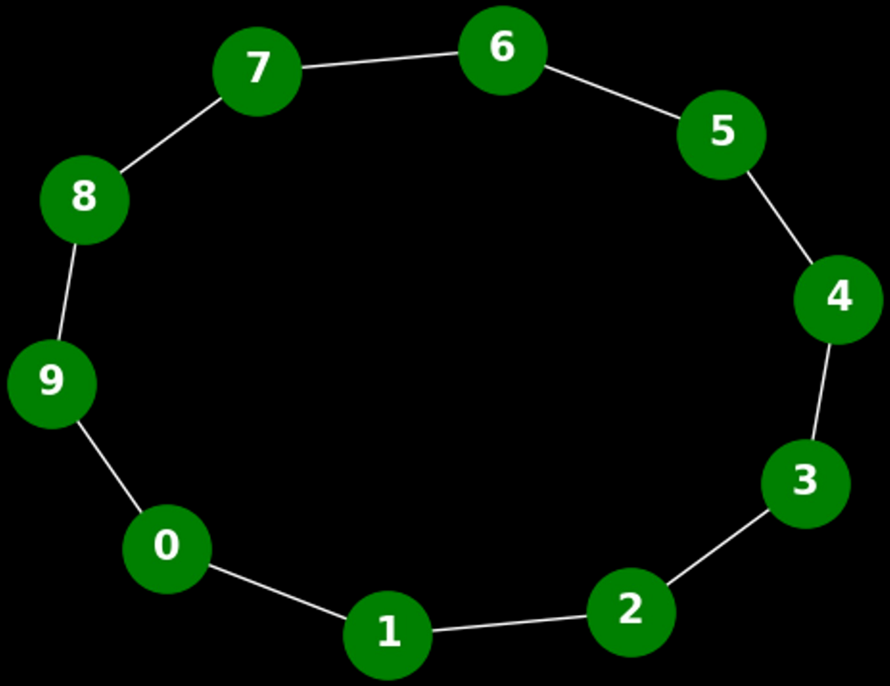


Mathematically…

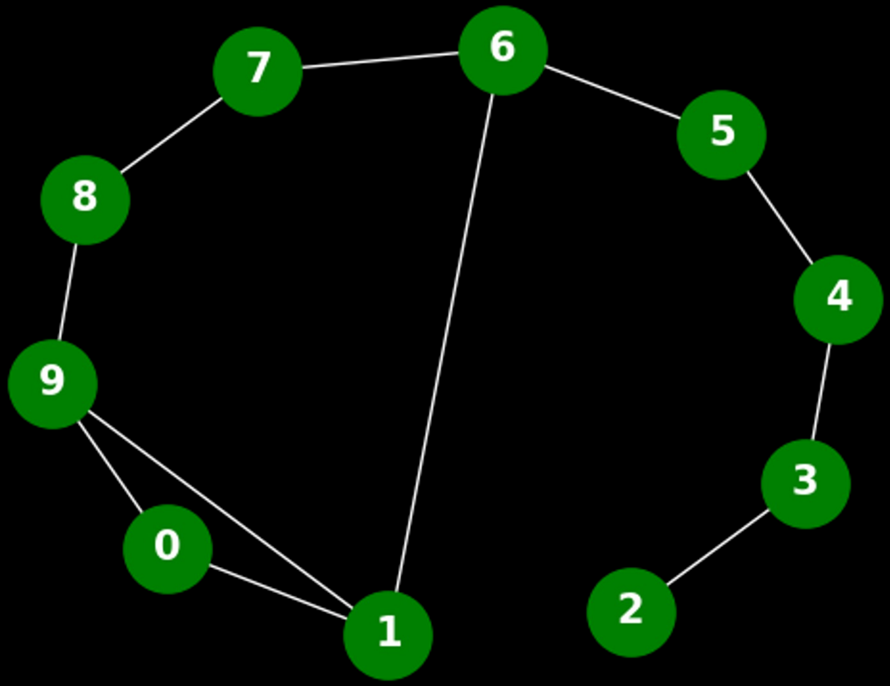
Building a network with a reduced sensitivity to these input perturbations, can be phrased as a constraint on the Lipschitz constant of the network, which should be "as small as possible" with respect to a suitable norm.

$$||\mathcal{N}_\theta(X+\delta) - \mathcal{N}_\theta(X)|| \leq c||\delta||$$

For graphs it is more complicated..



Attack to the connectivity of the Graph



e.g. A hacker adding or removing friendships on Facebook

Adversarial attacks on Graphs

$$F_* = F + \delta F, \quad \|\delta F\|_F \leq \varepsilon_1$$

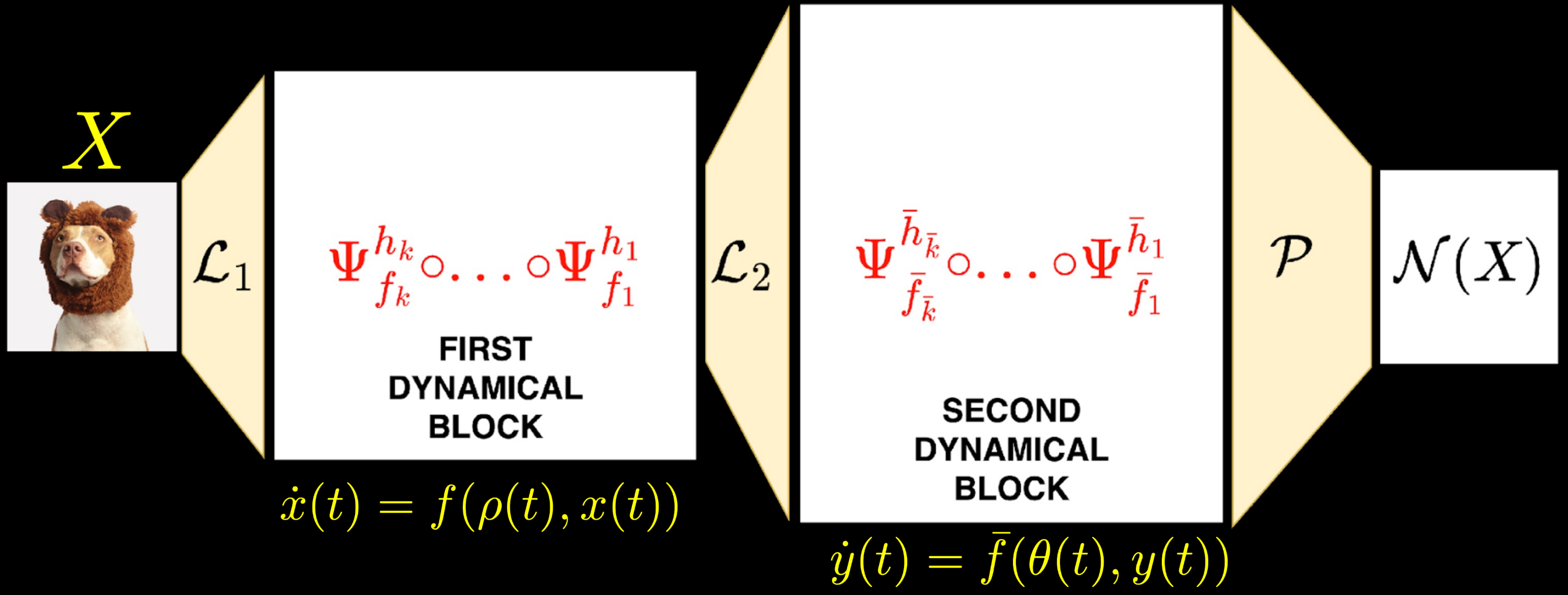
$$A_* = A + \delta A, \quad \sum_{i,j=1}^{n} |(\delta A)_{ij}| = \|\mathrm{vec}(\delta A)\|_1 \leq \varepsilon_2$$

Attacks do not break the properties of symmetry generally

Goal:

$$\mathrm{GNN}(F, A) \approx \mathrm{GNN}(F_*, A_*)$$

Dynamical systems-based neural networks



Example (ResNet) $\Psi_{f_i}^{h_i}(x) = x + h_i f_i(x)$

Our proposed architecture

Coupled Systems Graph Neural Network (CSGNN)

$$\begin{cases} \dot{F}(t) = -G(A(t))^T \sigma(G(A(t))F(t)W)W^T \\ \dot{A}(t) = \sigma(M(A(t))) \\ F(0) = F^{(0)},\ A(0) = A^{(0)} \end{cases}$$

And the solution is approximated with Explicit Euler steps of "small-enough" step size to obtain the neural network.

Some details on the functions

Graph Gradient Operator

$$\begin{aligned} (G(A)F)_{ijk} &= A_{ij}\left(F_{ik} - F_{jk}\right), & i,j &= 1,\cdots,n \\ & & k &= 1,\cdots,c \\ \left(G(A)^T O\right)_{ik} &= \sum_{j=1}^{n}\left(A_{ij}O_{ijk} - A_{ji}O_{jik}\right), & i &= 1,\cdots,n, \\ & & k &= 1,\cdots,c \end{aligned}$$

Linear Equivariant Map

$$\begin{aligned} M(A) = k_1 A + k_2 \operatorname{diag}(\operatorname{diag}(A)) + \frac{k_3}{2n}\left(A\mathbf{1}_n\mathbf{1}_n^T + \mathbf{1}_n\mathbf{1}_n^T A\right) + k_4 \operatorname{diag}\left(A\mathbf{1}_n\right) \\ + \frac{k_5}{n^2}\left(\mathbf{1}_n^T A\mathbf{1}_n\right)\mathbf{1}_n\mathbf{1}_n^T + \frac{k_6}{n}\left(\mathbf{1}_n^T A\mathbf{1}_n\right) I_n + \frac{k_7}{n^2}\left(\mathbf{1}_n^T \operatorname{diag}(A)\right)\mathbf{1}_n\mathbf{1}_n^T \\ + \frac{k_8}{n}\left(\mathbf{1}_n^T \operatorname{diag}(A)\right) I_n + \frac{k_9}{2n}\left(\operatorname{diag}(A)\mathbf{1}_n^T + \mathbf{1}_n(\operatorname{diag}(A))^T\right) \end{aligned}$$

$$M(PAP^T) = PM(A)P^T,\; M(A) = M(A)^T$$

Non-expansivity of the system

(Result based on “Contractive Systems with Inputs”, Eduardo D. Sontag)

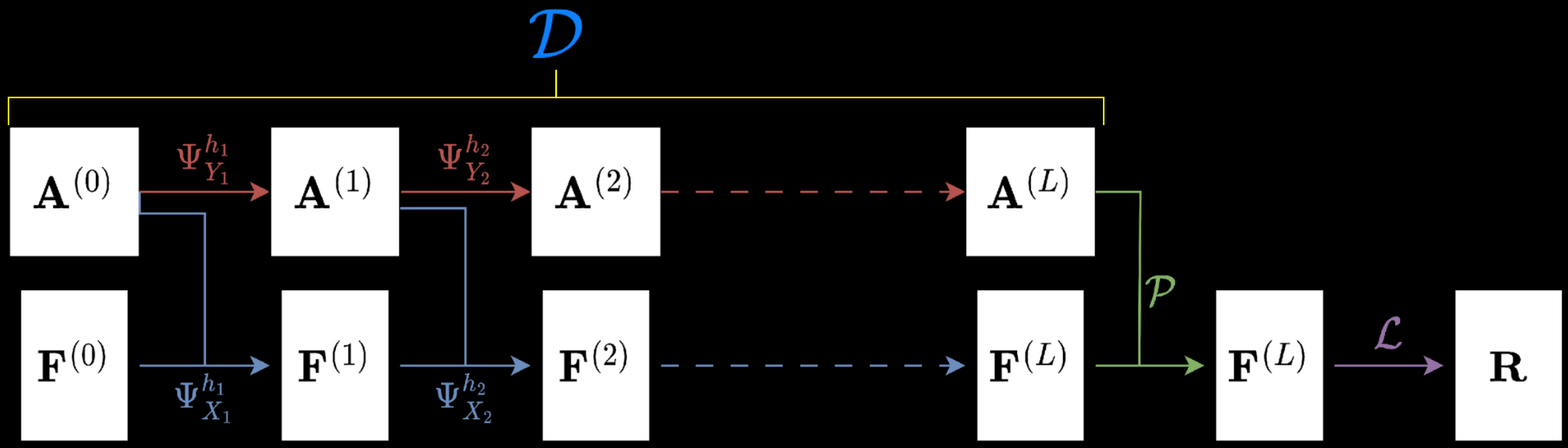
If $\sigma(x) = \max\{ax, x\},\ a \in (0,1)$, then for a suitable choice of the coefficient k_1 , the two individual systems are contractive, i.e.

$$\begin{aligned} \|F(t) - F_*(t)\|_F &\leq e^{-\nu_1 t} \left\| F^{(0)} - F_*^{(0)} \right\|_F, \nu_1 > 0, t \geq 0 \\ \|\mathrm{vec}(A(t)) - \mathrm{vec}\left(A_*(t)\right)\|_1 &\leq e^{-\nu_2 t} \left\| \mathrm{vec}\left(A^{(0)}\right) - \mathrm{vec}\left(A_*^{(0)}\right) \right\|_1, \nu_2 > 0, t \geq 0 \end{aligned}$$

and there is a pair of constants $m_1, m_2 > 0$ such that the coupled system satisfies

$$\begin{aligned} & m_1 \|F(t) - F_*(t)\|_F + m_2 \|\mathrm{vec}(A(t)) - \mathrm{vec}\left(A_*(t)\right)\|_1 \\ \leq & m_1 \left\| F^{(0)} - F_*^{(0)} \right\|_F + m_2 \left\| \mathrm{vec}\left(A^{(0)}\right) - \mathrm{vec}\left(A_*^{(0)}\right) \right\|_1 \end{aligned}$$

CSGNN



$$\left(F^{(0)}, A^{(0)}\right) := \left(\mathcal{K}\left(F_*\right), A_*\right)$$

$$\Psi^{h_i}_{X_i}(F, A) = F - h_i G(A)^T \sigma\left(G(A) F W_i\right) W_i^T$$

$$\Psi^{h_i}_{Y_i}(A) = A + h_i \sigma\left(M_i(A)\right)$$

Focus on the feature updates

If $\sigma : \mathbb{R} \to \mathbb{R}$ is a non-decreasing 1-Lipschitz function, then the explicit Euler update is non-expansive in the Frobenius norm for a small enough step-size, i.e.

$$\left\| \Psi_{X_i}^{h_i}(F + \delta F, A) - \Psi_{X_i}^{h_i}(F, A) \right\|_F \leq \|\delta F\|_F, \quad \delta F \in \mathbb{R}^{n \times c}$$

Focus on the adjacency updates

If $\sigma : \mathbb{R} \to \mathbb{R}$ is a non-decreasing 1-Lipschitz function, then the explicit Euler update is non-expansive in the vectorized 1-norm for a small enough step-size when

$$k_1 = \left(\alpha - \sum_{i=2}^{9} |k_i|\right), \ \alpha \leq 0.$$

This means that:

$$\left\|\mathrm{vec}(\Psi_{Y_i}^{h_i}(A + \delta A)) - \mathrm{vec}(\Psi_{Y_i}^{h_i}(A))\right\|_1 \leq \|\mathrm{vec}(\delta A)\|_1,$$

$$\delta A \in \mathbb{R}^{n \times n}$$

Robustness of the network

If the assumptions of the two previous theorems hold, and

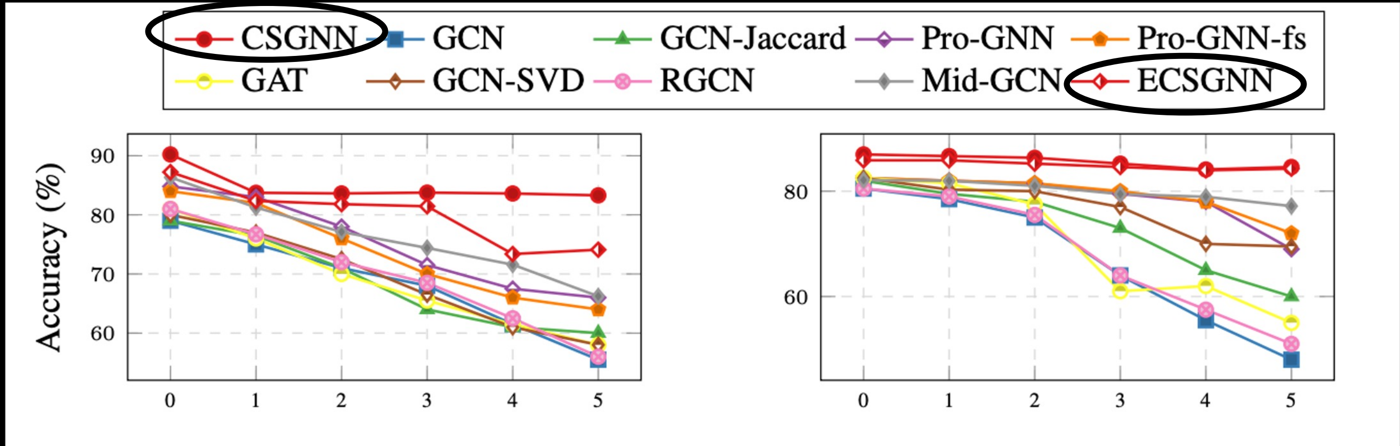
$$F_* = F + \delta F, \quad \|\delta F\|_F \leq \varepsilon_1$$
$$A_* = A + \delta A, \quad \|\operatorname{vec}(\delta A)\|_1 \leq \varepsilon_2$$

it follows

$$\begin{aligned} &\left\|\operatorname{vec}\left(A^{(L)}\right) - \operatorname{vec}\left(A_*^{(L)}\right)\right\|_1 + \left\|F^{(L)} - F_*^{(L)}\right\|_F \\ &\leq \varepsilon_1 + \varepsilon_2 \left(1 + \sum_{i=1}^{L} \operatorname{Lip}\left(X_{i,F^{(i-1)}}\right) h_i\right) \\ &=: \varepsilon_1 + c\left(h_1, \ldots, h_L\right) \varepsilon_2. \end{aligned}$$

Some experimental results

Method	Cora nettack	Cora metattack	Cora random	Citeseer nettack	Citeseer metattack	Citeseer random
$\text{CSGNN}_{\text{noAdj}}$	81.90	70.25	77.19	82.20	70.17	71.28
CSGNN	83.29	74.46	78.38	84.60	72.94	72.70

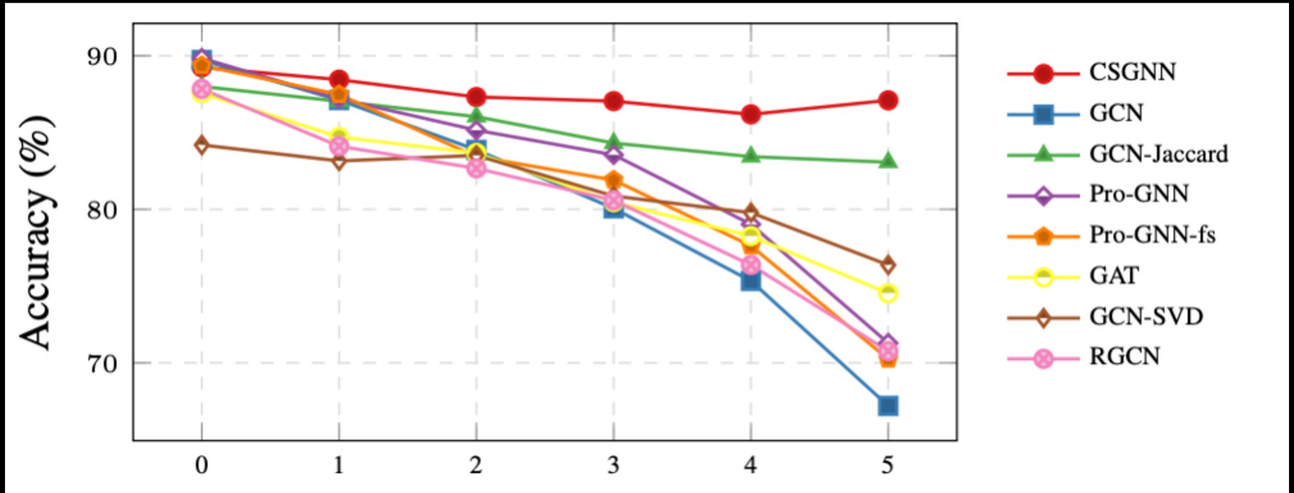


(a) Cora (b) Citeseer

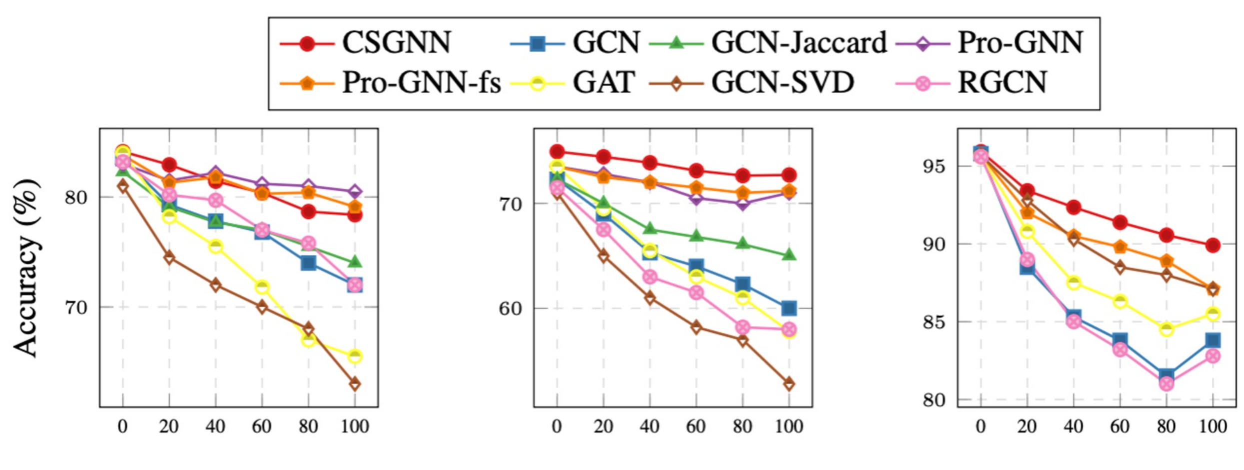
We target the nodes with degree at least 10 and flip few of their incident edges

Node classification accuracy (%) of ECSGNN and other baselines, under a targeted attack generated by nettack. The horizontal axis describes the number of perturbations per node.

Some experimental results



Classification accuracy for The Pubmed dataset using Nettack as attack method.



(a) Cora (b) Citeseer (c) Polblogs

The adjacency matrix is attacked by adding random fake edges, from 0% to 100% of the number of edges in the true one.

Thank you for the attention