Injecting Hamiltonian Architectural Bias into Deep Graph Networks

AI4Science Talk by Simon Heilig[1] & Alessio Gravina[2]

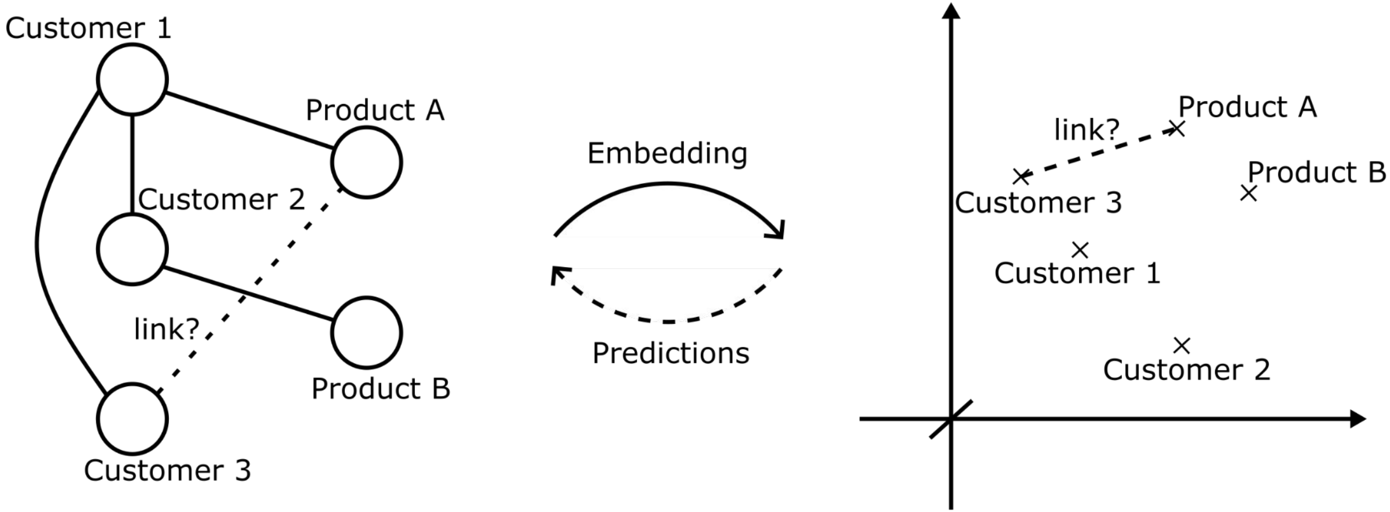
[1] ELLIS Ph.D. Student, Ruhr University Bochum
[2] Ph.D. Candidate, University of Pisa

Outline

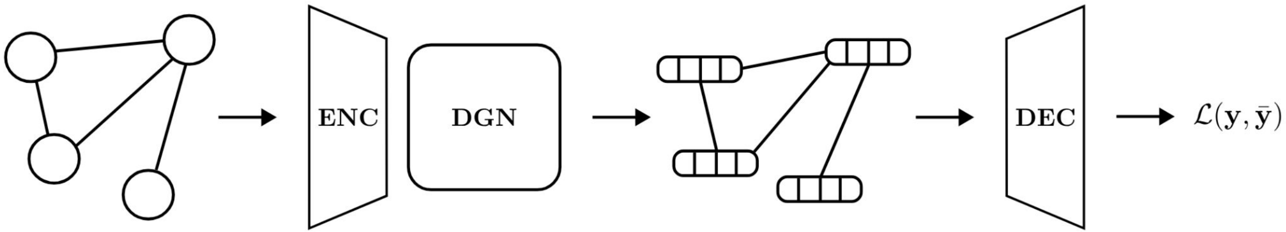
- Background
  - ML on Graphs
  - Message Passing Neural Networks
  - Neural ODEs and Hamiltonian Dynamics
- Contribution
  - (Port-)Hamiltonian Graph Neural ODE
  - Theoretical Properties
- Experimental Validation
  - Graph Transfer
  - Graph Property Prediction
  - Long Range Graph Benchmark

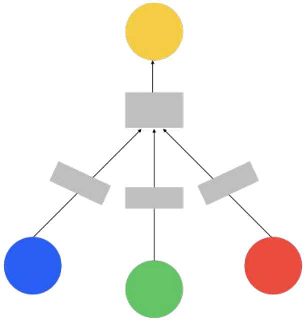
Background

Motivation: ML on Graphs



Message Passing Neural Networks





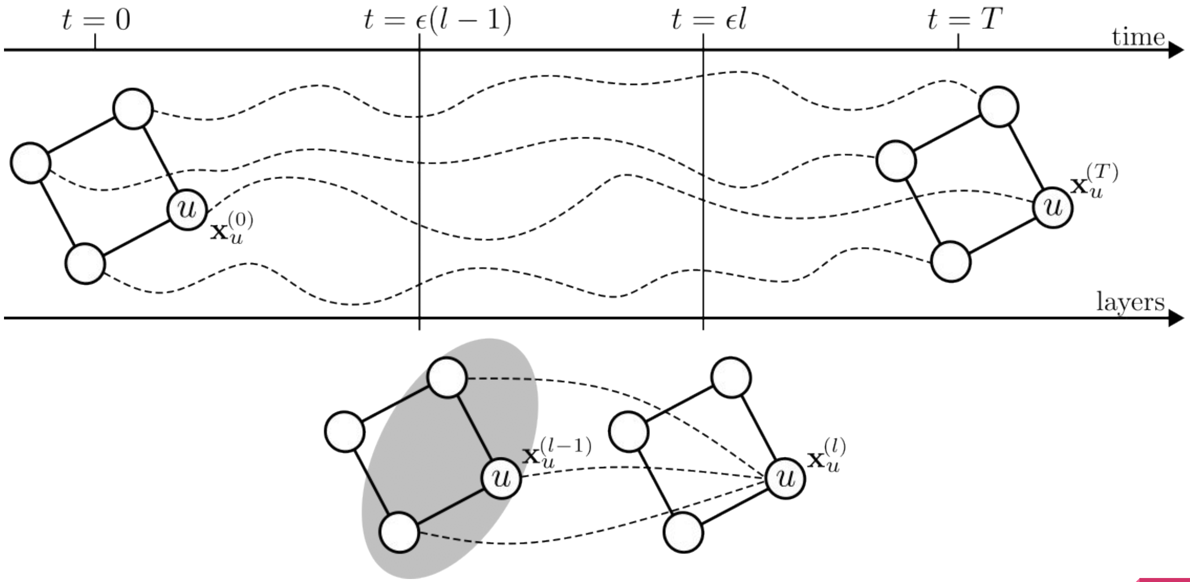
$$\mathbf{x}_u^{(l+1)} = \mathrm{UP}^{(l+1)}\left(\mathbf{x}_u^{(l)}, \mathrm{AGG}^{(l+1)}\left(\left\{\mathbf{x}_v^{(l)} \, : \, v \in \mathcal{N}_u\right\}\right)\right)$$

$$\mathbf{x}_u^{(l+1)} = \sigma\left(\mathbf{W}_{\text{self}}^{(l+1)}\mathbf{x}_u^{(l)} + \sum_{v\in\mathcal{N}_u} \mathbf{W}_{\text{msg}}^{(l+1)}\mathbf{x}_v^{(l)}\right)$$

$$\mathbf{X}^{(l+1)} = \mathbf{X}^{(l)} + \sigma\left(-\mathbf{X}^{(l)}\mathbf{W}_{\text{self}}^{(l+1)} + \mathbf{A}_{\text{agg}}\mathbf{X}^{(l)}\mathbf{W}_{\text{msg}}^{(l+1)} - \mathbf{X}^{(0)}\mathbf{W}_{\text{skip}}^{(l+1)}\right)$$

Ref: [1]

Graph Coupled Neural ODEs

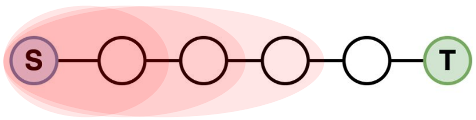


$$\frac{\mathrm{d}\mathbf{x}_u(t)}{\mathrm{d}t} = \sigma\left(\mathbf{W}_{\text{self}}(t)\mathbf{x}_u(t) + \Phi_{\mathcal{G}}(\{\mathbf{x}_v(t) : v \in \mathcal{N}_u\}, t) + \mathbf{b}(t)\right)$$

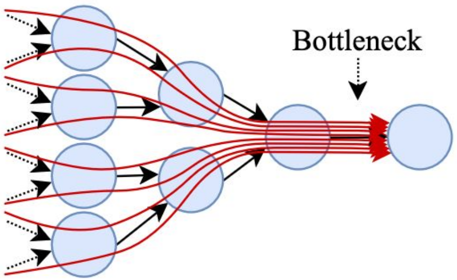
Ref: [4]

The Long-Range Regime

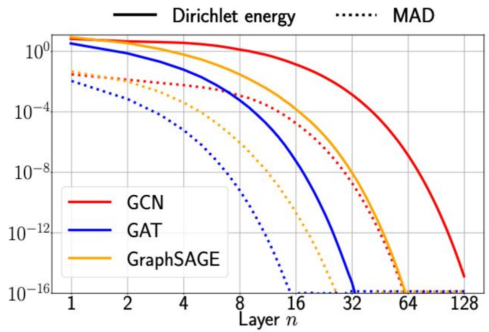
Under Reaching



Over Squashing



Over Smoothing



Hamiltonian Dynamics

Example (Equations of Motion)

$$H(\mathbf{p},\mathbf{q}) = T(\mathbf{p}) + U(\mathbf{q}), \quad \text{with } T = \frac{1}{2}\sum_{i=1,\dots,n}\frac{\|\mathbf{p}_i\|^2}{m_i}, U = -\sum_{1\leq i<j\leq n}\frac{Gm_im_j}{\|\mathbf{q}_j-\mathbf{q}_i\|}, \mathbf{p},\mathbf{q}\in\mathbb{R}^{3n}$$

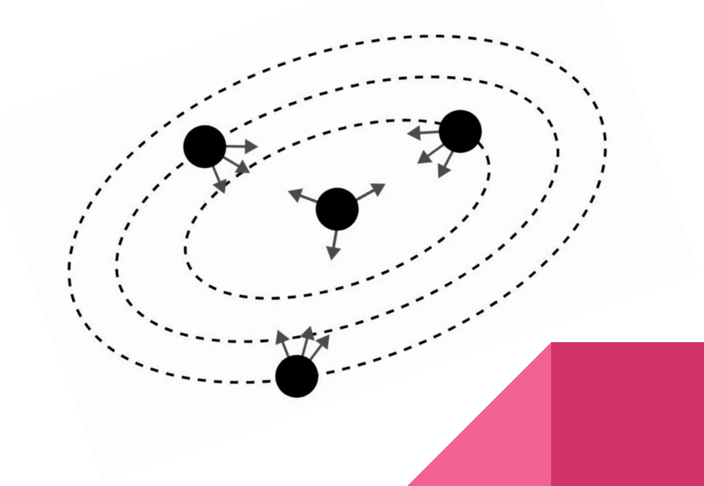
Definition (Hamiltonian Dynamics)

$$\frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = -\frac{\partial H}{\partial \mathbf{q}}(\mathbf{p},\mathbf{q}), \quad \frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} = \frac{\partial H}{\partial \mathbf{p}}(\mathbf{p},\mathbf{q})$$

$$\Leftrightarrow$$

$$\frac{\mathrm{d}\mathbf{y}(t)}{\mathrm{d}t} = \mathcal{J}\nabla_{\mathbf{y}}H(\mathbf{y}(t)),$$

where $\mathbf{y} = (\mathbf{p},\mathbf{q})^\top \in \mathbb{R}^{2d}$ and $\mathcal{J} = \begin{pmatrix} \mathbf{0} & -\mathbf{I}_d \\ \mathbf{I}_d & \mathbf{0} \end{pmatrix}$



Ref: [5]

(Port-)Hamiltonian DGN

Hamiltonian Inspired Message Passing

Step 1)

$$\mathbf{y} := \begin{pmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_n \\ \mathbf{q}_1 \\ \vdots \\ \mathbf{q}_n \end{pmatrix} = \begin{pmatrix} \mathbf{p} \\ \mathbf{q} \end{pmatrix} \in \mathbb{R}^{nd}, \text{with } \begin{pmatrix} \mathbf{p}_u \\ \mathbf{q}_u \end{pmatrix} := \mathbf{x}_u \in \mathbb{R}^d, \text{ and } H_{\mathcal{G}}(\mathbf{y}) := \sum_{u \in \mathcal{V}}^{n} \tilde{\sigma}(\mathbf{W}\mathbf{x}_u + \Phi_{\mathcal{G}}(\mathbf{y}, \mathcal{N}_u) + \mathbf{b})^\top \mathbf{1}_d$$

Step 2)

$$\frac{\mathrm{d}\mathbf{y}(t)}{\mathrm{d}t} = \mathcal{J}\nabla_{\mathbf{y}} H_{\mathcal{G}}(\mathbf{y}(t)) \Leftrightarrow \frac{\mathrm{d}\mathbf{p}(t)}{\mathrm{d}t} = -\nabla_{\mathbf{q}} H_{\mathcal{G}}(\mathbf{y}(t)), \frac{\mathrm{d}\mathbf{q}(t)}{\mathrm{d}t} = \nabla_{\mathbf{p}} H_{\mathcal{G}}(\mathbf{y}(t))$$

$$\Leftrightarrow$$

$$\frac{\mathrm{d}\mathbf{x}_u(t)}{\mathrm{d}t} = \begin{pmatrix} -\nabla_{\mathbf{q}_u} H_{\mathcal{G}}(\mathbf{y}(t)) \\ \nabla_{\mathbf{p}_u} H_{\mathcal{G}}(\mathbf{y}(t)) \end{pmatrix} = \mathcal{J}_u \nabla_{\mathbf{x}_u} H_{\mathcal{G}}(\mathbf{y}(t)) \quad , \forall\, u \in \mathcal{V}$$

$$\frac{\mathrm{d}H_{\mathcal{G}}}{\mathrm{d}t} = 0$$

Continuous and Discrete Perspective

The Continuous Dynamic

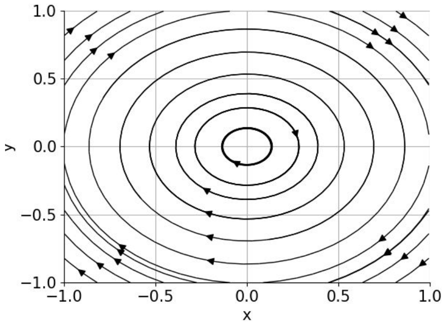
$$\frac{\mathrm{d}\mathbf{x}_u(t)}{\mathrm{d}t} = \mathcal{J}_u \nabla_{\mathbf{x}_u} H_{\mathcal{G}}(\mathbf{y}(t)) = \mathcal{J}_u \left[\mathbf{W}^\top \sigma(\mathbf{W}\mathbf{x}_u(t) + \Phi_{\mathcal{G}}(\mathbf{y}, \mathcal{N}_u) + \mathbf{b}) + \sum_{v \in \mathcal{N}_u \cup \{u\}} \frac{\partial \Phi_{\mathcal{G}}(\mathbf{y}, \mathcal{N}_v)}{\partial \mathbf{x}_u(t)}^\top \sigma(\mathbf{W}\mathbf{x}_v(t) + \Phi_{\mathcal{G}}(\mathbf{y}, \mathcal{N}_v) + \mathbf{b}) \right]$$

Symplectic Integration

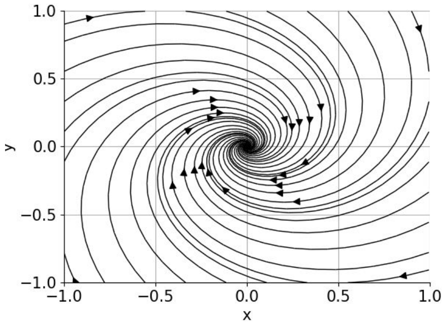
$$\begin{pmatrix} \mathbf{p}_u^{(n+1)} \\ \mathbf{q}_u^{(n+1)} \end{pmatrix} = \begin{pmatrix} \mathbf{p}_u^{(n)} \\ \mathbf{q}_u^{(n)} \end{pmatrix} + \epsilon \mathcal{J}_u \begin{pmatrix} \nabla_{\mathbf{p}_u} H_{\mathcal{G}}(\mathbf{p}^{(n+1)}, \mathbf{q}^{(n)}) \\ \nabla_{\mathbf{q}_u} H_{\mathcal{G}}(\mathbf{p}^{(n+1)}, \mathbf{q}^{(n)}) \end{pmatrix} \quad , \forall u \in \mathcal{V}$$

Assumptions on $\mathbf{W}$ and $\mathbf{V}$ for $\Phi_{\mathcal{G}}((\mathbf{p}^{(n+1)}, \mathbf{q}^{(n)}), \mathcal{N}) = \sum_{i \in \mathcal{N}} \mathbf{V} \begin{pmatrix} \mathbf{p}_i^{(n+1)} \\ \mathbf{q}_i^{(n)} \end{pmatrix}$: $\mathbf{W} = \begin{pmatrix} \mathbf{W}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_q \end{pmatrix}$ and $\mathbf{V} = \begin{pmatrix} \mathbf{V}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_q \end{pmatrix}$

Conservation and Dissipation



vs.

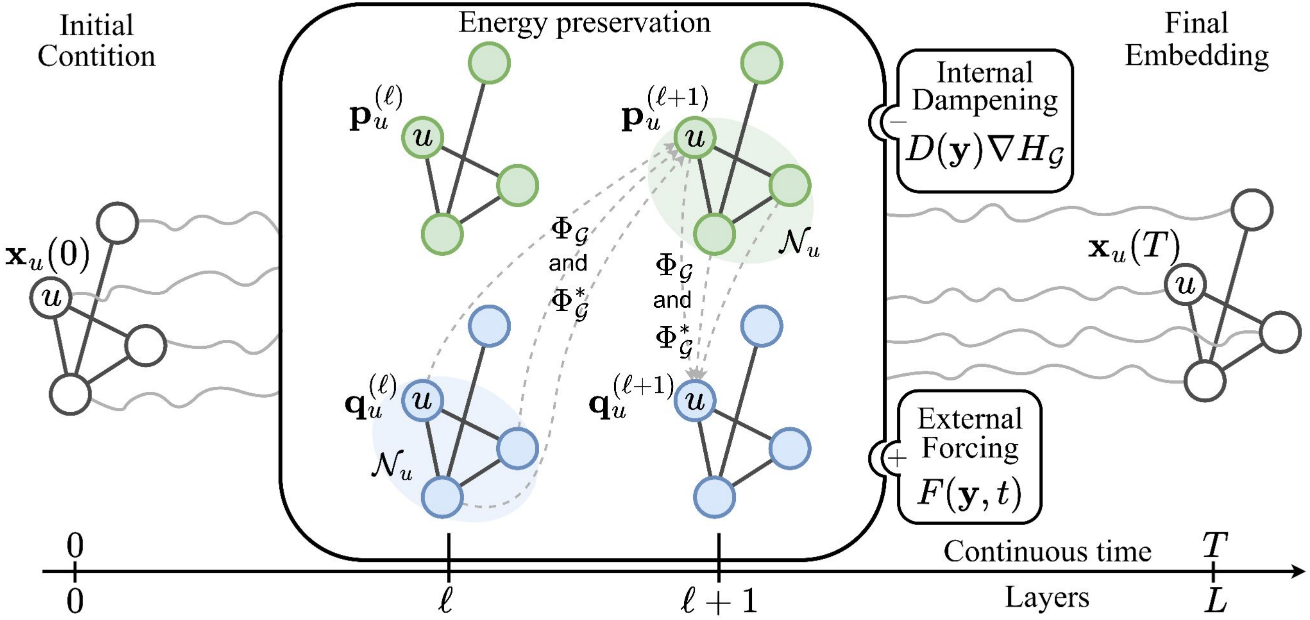


Port-Hamiltonian Dynamics

$$\frac{\mathrm{d}\mathbf{x}_u(t)}{\mathrm{d}t} = \left(\mathcal{J}_u - \begin{bmatrix} D(\mathbf{q}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\right) \nabla_{\mathbf{x}_u} H_{\mathcal{G}}(\mathbf{y}) + \begin{bmatrix} F(\mathbf{q}, t) \\ \mathbf{0} \end{bmatrix}, \quad \forall u \in \mathcal{V}.$$

Ref: [2]

Big Picture: Port-Hamiltonian DGN



Stability Analysis

Lemma (Eigenvalues of the Jacobian)

The local dynamics possess an Jacobian with eigenvalues purely on the imaginary axis, i.e.

$$\mathrm{Re}\left(\lambda_i\left(\frac{\partial}{\partial \mathbf{x}_u}\mathcal{J}_u \nabla_{\mathbf{x}_u} H_{\mathcal{G}}(\mathbf{y}(t))\right)\right) = 0 \quad , \forall i$$

Lemma (Divergence-Free Hamiltonian Vector Field)

The vector field defined by the local dynamics is divergence-free everywhere, which means:

$$\nabla \cdot \mathcal{J}_u \nabla_{\mathbf{x}_u} H_{\mathcal{G}}(\mathbf{y}(t)) = 0, \text{ everywhere}$$

Hence, no sources or sinks can exist in the nonlinear dynamic.

→ Allows for long-range propagation!

Sensitivity Analysis I

Theorem (Lower Bound)

Given a Hamiltonian graph ODE, the BSM for each node can not vanish. In particular, for any sub-multiplicative matrix norm $\|\cdot\|$, the lower bound is:

$$\left\| \frac{\partial \mathbf{x}_u(T)}{\partial \mathbf{x}_u(T-t)} \right\| \geq 1, \forall t \in [0, T].$$

Given a discrete Hamiltonian graph ODE obtained by a symplectic discretization scheme, the BSM can not vanish:

$$\left\| \frac{\partial \mathbf{x}_u^{(L)}}{\partial \mathbf{x}_u^{(L-j)}} \right\| \geq 1, \forall j = 0, \dots, L-1.$$

Ref: [3]

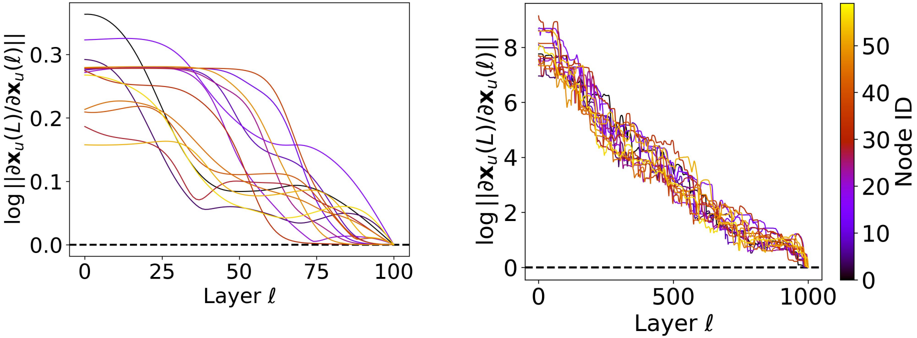
Sensitivity Analysis II

Theorem (Upper Bound)

Given a Hamiltonian graph ODE, with a nonlinearity whose derivative is bounded globally, i.e., $\exists M > 0, |\sigma'(x)| \leq M$*, and a neighborhood coupling of the form* $\Phi_{\mathcal{G}} = \sum_{v \in \mathcal{N}_u} \mathbf{V}\mathbf{x}_v$*, the BSM can be bounded above as:*

$$\left\| \frac{\partial \mathbf{x}_u(T)}{\partial \mathbf{x}_u(T-t)} \right\|_2 \leq \sqrt{d}\, \mathit{exp}(QT), \quad \forall t \in [0, T]$$

Where $Q = \sqrt{d}\, M \|\mathbf{W}\|_2^2 + \sqrt{d}\, M\, \mathit{max}_{i \in [n]} |\mathcal{N}_i| \|\mathbf{V}\|_2^2$.



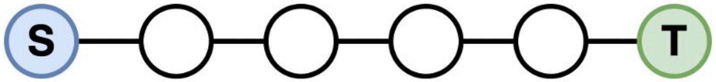
Experiments

Experimental outline

- Assess the efficacy in **preserving long-range information** between nodes
  - Graph transfer tasks
  - Graph property prediction tasks
  - Long Range Graph Benchmark

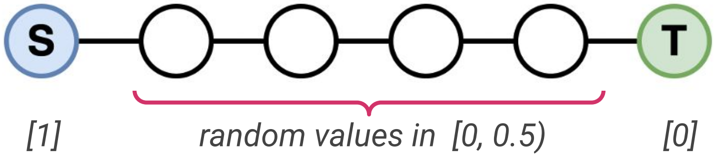
Graph Transfer task

- Propagate a label from a source to a target node



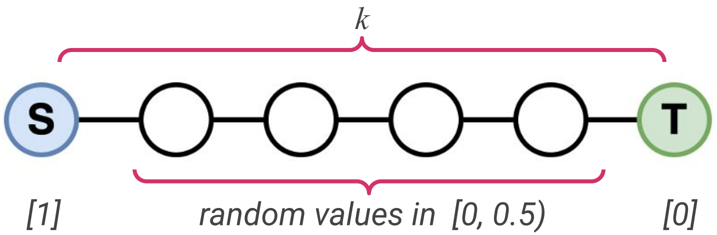
Graph Transfer task

- Propagate a label from a source to a target node



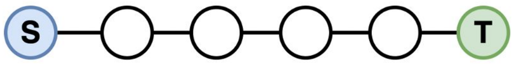
Graph Transfer task

- Propagate a label from a source to a target node
    - Increasing distance k

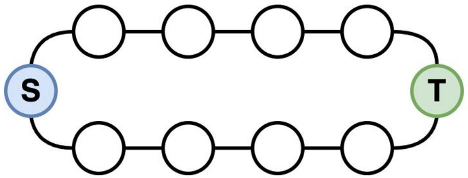


Graph Transfer task

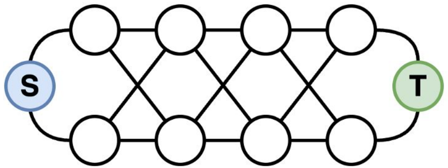
- Propagate a label from a source to a target node
  - Increasing distance k
  - Multiple graph topologies



Line graph



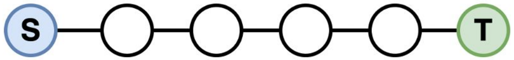
Ring graph



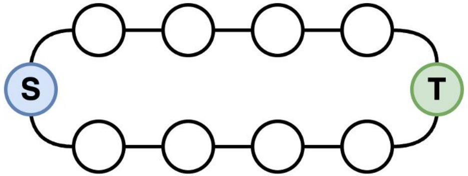
Crossed-Ring graph

Graph Transfer task

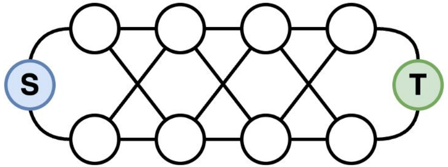
- Propagate a label from a source to a target node
  - Increasing distance k
  - Multiple graph topologies
  - MSE as metric



Line graph

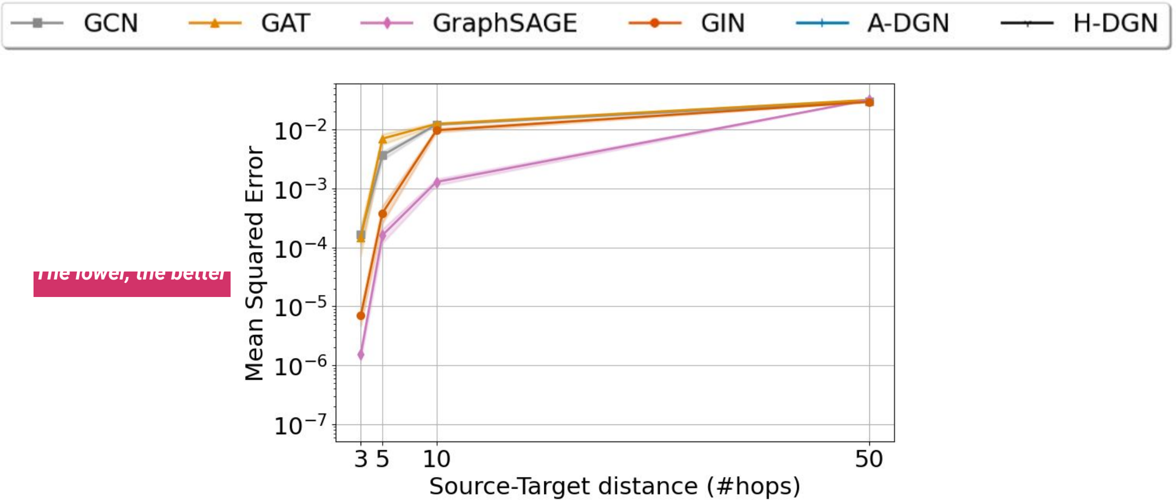


Ring graph



Crossed-Ring graph

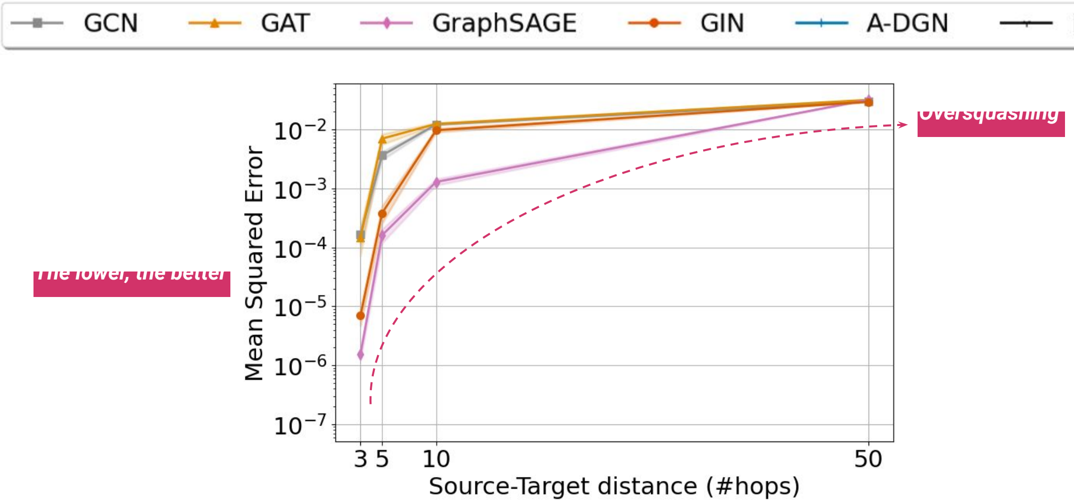
Graph Transfer task



The lower, the better

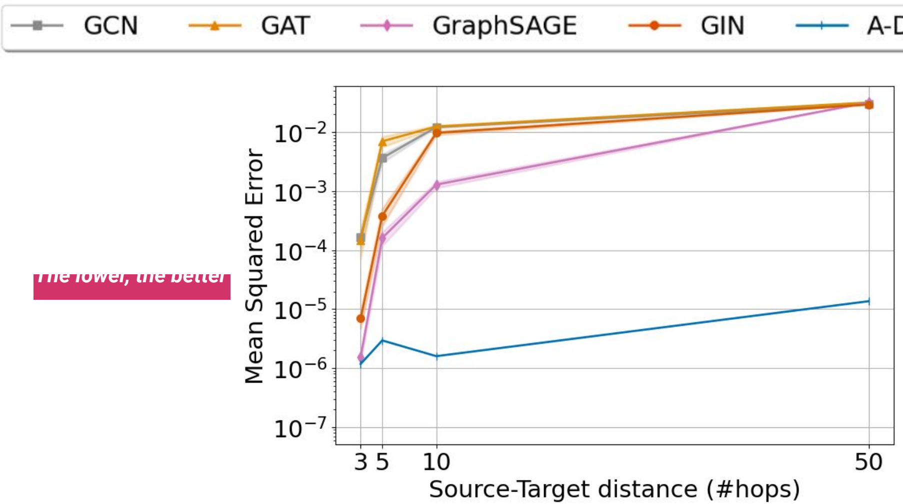
Line graph

Graph Transfer task



Line graph

Graph Transfer task



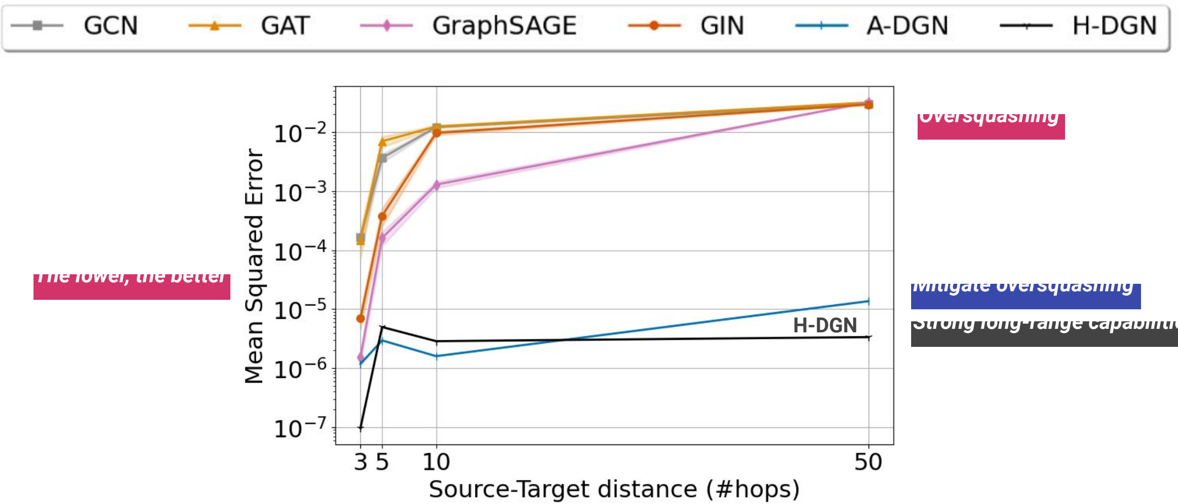
The lower, the better

Oversquashing

Mitigate oversquashing

Line graph

Graph Transfer task



Oversquashing

The lower, the better

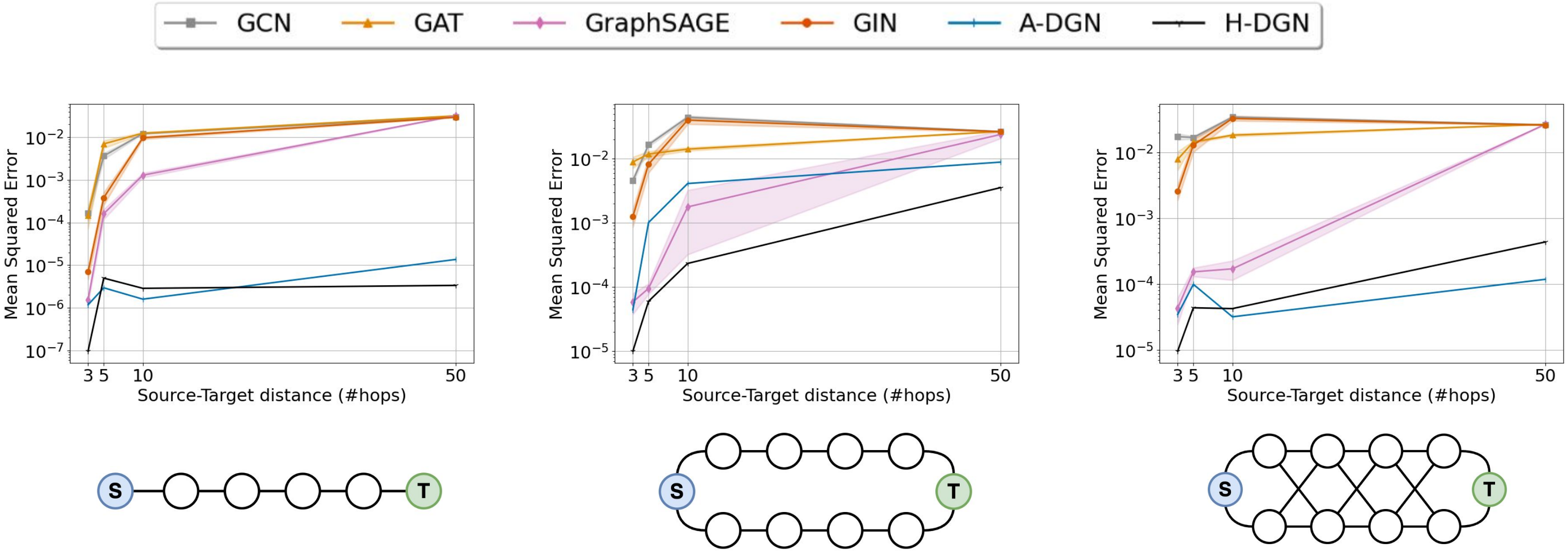
Mitigate oversquashing

Strong long-range capabilities

Line graph

Graph Transfer task

The lower, the better

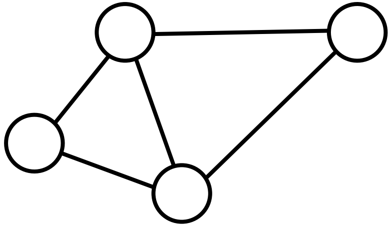


Graph Property Prediction

- Introduced in [4]
- Predict 3 graph properties

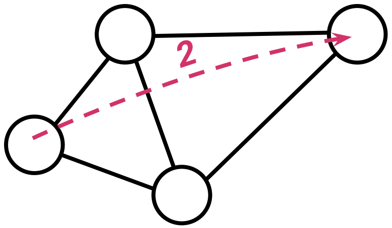
Graph Property Prediction

- Introduced in [4]
- Predict 3 graph properties
    - Graph diameter



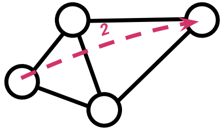
Graph Property Prediction

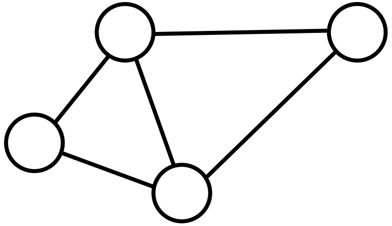
- Introduced in [4]
- Predict 3 graph properties
  - Graph diameter



Graph Property Prediction

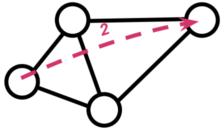
- Introduced in [4]
- Predict 3 graph properties
  - Graph diameter
  - Single source shortest path (SSSP)

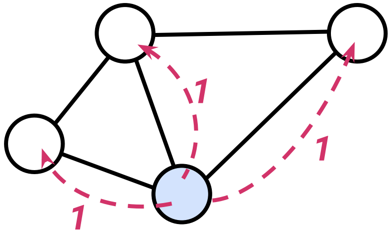




Graph Property Prediction

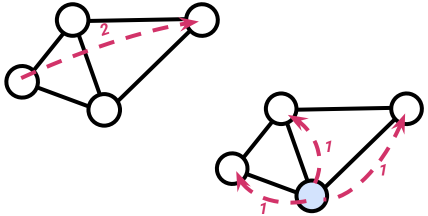
- Introduced in [4]
- Predict 3 graph properties
  - Graph diameter
  - Single source shortest path (SSSP)

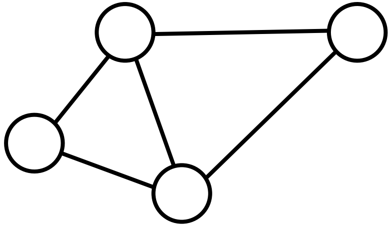




Graph Property Prediction

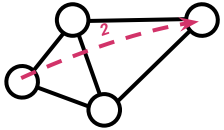
- Introduced in [4]
- Predict 3 graph properties
  - Graph diameter
  - Single source shortest path (SSSP)
  - Node eccentricity

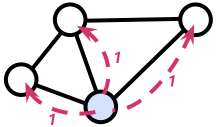


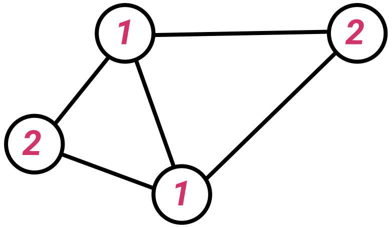


Graph Property Prediction

- Introduced in [4]
- Predict 3 graph properties
  - Graph diameter
  - Single source shortest path (SSSP)
  - Node eccentricity







Graph Property Prediction

Table 1: Mean test log_{10}(MSE) and std average over 4 training seeds on the Graph Property Prediction. Our methods and DE-DGN baselines are implemented with weight sharing. The **first**, **second**, and **third** best scores are colored.

The lower, the better

Model	Eccentricity	Diameter	SSSP
MPNNs			
GCN	$0.8468_{\pm 0.0028}$	$0.7424_{\pm 0.0466}$	$0.9499_{\pm 9.2 \cdot 10^{-5}}$
GAT	$0.7909_{\pm 0.0222}$	$0.8221_{\pm 0.0752}$	$0.6951_{\pm 0.1499}$
GraphSAGE	$0.7863_{\pm 0.0207}$	$0.8645_{\pm 0.0401}$	$0.2863_{\pm 0.1843}$
GIN	$0.9504_{\pm 0.0007}$	$0.6131_{\pm 0.0990}$	$-0.5408_{\pm 0.4193}$
GCNII	$0.7640_{\pm 0.0355}$	$0.5287_{\pm 0.0570}$	$-1.1329_{\pm 0.0135}$
DE-DGNs			
DGC	$0.8261_{\pm 0.0032}$	$0.6028_{\pm 0.0050}$	$-0.1483_{\pm 0.0231}$
GraphCON	$0.6833_{\pm 0.0074}$	$0.0964_{\pm 0.0620}$	$-1.3836_{\pm 0.0092}$
GRAND	$0.6602_{\pm 0.1393}$	$0.6715_{\pm 0.0490}$	$-0.0942_{\pm 0.3897}$
A-DGN	$\mathbf{0.4296}_{\pm 0.1003}$	$\mathbf{-0.5188}_{\pm 0.1812}$	$\mathbf{-3.2417}_{\pm 0.0751}$
Ours			
H-DGN	$\mathbf{-0.7248}_{\pm 0.1068}$	$\mathbf{-0.5473}_{\pm 0.1074}$	$\mathbf{-3.0467}_{\pm 0.1615}$
PH-DGN	$\mathbf{-0.9348}_{\pm 0.2097}$	$\mathbf{-0.5385}_{\pm 0.0187}$	$\mathbf{-4.2993}_{\pm 0.0721}$

Graph Property Prediction

Table 1: Mean test log_{10}(MSE) and std average over 4 training seeds on the Graph Property Prediction. Our methods and DE-DGN baselines are implemented with weight sharing. The **first**, **second**, and **third** best scores are colored.

The lower, the better

Model	Eccentricity	Diameter	SSSP
MPNNs			
GCN	$0.8468_{\pm 0.0028}$	$0.7424_{\pm 0.0466}$	$0.9499_{\pm 9.2 \cdot 10^{-5}}$
GAT	$0.7909_{\pm 0.0222}$	$0.8221_{\pm 0.0752}$	$0.6951_{\pm 0.1499}$
GraphSAGE	$0.7863_{\pm 0.0207}$	$0.8645_{\pm 0.0401}$	$0.2863_{\pm 0.1843}$
GIN	$0.9504_{\pm 0.0007}$	$0.6131_{\pm 0.0990}$	$-0.5408_{\pm 0.4193}$
GCNII	$0.7640_{\pm 0.0355}$	$0.5287_{\pm 0.0570}$	$-1.1329_{\pm 0.0135}$
DE-DGNs			
DGC	$0.8261_{\pm 0.0032}$	$0.6028_{\pm 0.0050}$	$-0.1483_{\pm 0.0231}$
GraphCON	$0.6833_{\pm 0.0074}$	$0.0964_{\pm 0.0620}$	$-1.3836_{\pm 0.0092}$
GRAND	$0.6602_{\pm 0.1393}$	$0.6715_{\pm 0.0490}$	$-0.0942_{\pm 0.3897}$
A-DGN	$\mathbf{0.4296}_{\pm 0.1003}$	$\mathbf{-0.5188}_{\pm 0.1812}$	$\mathbf{-3.2417}_{\pm 0.0751}$
Ours			
H-DGN	$\mathbf{-0.7248}_{\pm 0.1068}$	$\mathbf{-0.5473}_{\pm 0.1074}$	$\mathbf{-3.0467}_{\pm 0.1615}$
PH-DGN	$\mathbf{-0.9348}_{\pm 0.2097}$	$\mathbf{-0.5385}_{\pm 0.0187}$	$\mathbf{-4.2993}_{\pm 0.0721}$

0.33 log(MSE) better than the best baseline on average

Graph Property Prediction

Table 1: Mean test log_{10}(MSE) and std average over 4 training seeds on the Graph Property Prediction. Our methods and DE-DGN baselines are implemented with weight sharing. The **first**, **second**, and **third** best scores are colored.

The lower, the better

Model	Eccentricity	Diameter	SSSP
MPNNs			
GCN	$0.8468_{\pm 0.0028}$	$0.7424_{\pm 0.0466}$	$0.9499_{\pm 9.2 \cdot 10^{-5}}$
GAT	$0.7909_{\pm 0.0222}$	$0.8221_{\pm 0.0752}$	$0.6951_{\pm 0.1499}$
GraphSAGE	$0.7863_{\pm 0.0207}$	$0.8645_{\pm 0.0401}$	$0.2863_{\pm 0.1843}$
GIN	$0.9504_{\pm 0.0007}$	$0.6131_{\pm 0.0990}$	$-0.5408_{\pm 0.4193}$
GCNII	$0.7640_{\pm 0.0355}$	$0.5287_{\pm 0.0570}$	$-1.1329_{\pm 0.0135}$
DE-DGNs			
DGC	$0.8261_{\pm 0.0032}$	$0.6028_{\pm 0.0050}$	$-0.1483_{\pm 0.0231}$
GraphCON	$0.6833_{\pm 0.0074}$	$0.0964_{\pm 0.0620}$	$-1.3836_{\pm 0.0092}$
GRAND	$0.6602_{\pm 0.1393}$	$0.6715_{\pm 0.0490}$	$-0.0942_{\pm 0.3897}$
A-DGN	$\mathbf{0.4296}_{\pm 0.1003}$	$\mathbf{-0.5188}_{\pm 0.1812}$	$\mathbf{-3.2417}_{\pm 0.0751}$
Ours			
H-DGN	$\mathbf{-0.7248}_{\pm 0.1068}$	$\mathbf{-0.5473}_{\pm 0.1074}$	$\mathbf{-3.0467}_{\pm 0.1615}$
PH-DGN	$\mathbf{-0.9348}_{\pm 0.2097}$	$\mathbf{-0.5385}_{\pm 0.0187}$	$\mathbf{-4.2993}_{\pm 0.0721}$

0.33 log(MSE) better

0.81 log(MSE) better than the best baseline on average

Graph Property Prediction

Table 1: Mean test log_{10}(MSE) and std average over 4 training seeds on the Graph Property Prediction. Our methods and DE-DGN baselines are implemented with weight sharing. The **first**, **second**, and **third** best scores are colored.

The lower, the better

Model	Eccentricity	Diameter	SSSP
MPNNs			
GCN	$0.8468_{\pm 0.0028}$	$0.7424_{\pm 0.0466}$	$0.9499_{\pm 9.2 \cdot 10^{-5}}$
GAT	$0.7909_{\pm 0.0222}$	$0.8221_{\pm 0.0752}$	$0.6951_{\pm 0.1499}$
GraphSAGE	$0.7863_{\pm 0.0207}$	$0.8645_{\pm 0.0401}$	$0.2863_{\pm 0.1843}$
GIN	$0.9504_{\pm 0.0007}$	$0.6131_{\pm 0.0990}$	$-0.5408_{\pm 0.4193}$
GCNII	$0.7640_{\pm 0.0355}$	$0.5287_{\pm 0.0570}$	$-1.1329_{\pm 0.0135}$
DE-DGNs			
DGC	$0.8261_{\pm 0.0032}$	$0.6028_{\pm 0.0050}$	$-0.1483_{\pm 0.0231}$
GraphCON	$0.6833_{\pm 0.0074}$	$0.0964_{\pm 0.0620}$	$-1.3836_{\pm 0.0092}$
GRAND	$0.6602_{\pm 0.1393}$	$0.6715_{\pm 0.0490}$	$-0.0942_{\pm 0.3897}$
A-DGN	$\mathbf{0.4296}_{\pm 0.1003}$	$\mathbf{-0.5188}_{\pm 0.1812}$	$\mathbf{-3.2417}_{\pm 0.0751}$
Ours			
H-DGN	$\mathbf{-0.7248}_{\pm 0.1068}$	$\mathbf{-0.5473}_{\pm 0.1074}$	$\mathbf{-3.0467}_{\pm 0.1615}$
PH-DGN	$\mathbf{-0.9348}_{\pm 0.2097}$	$\mathbf{-0.5385}_{\pm 0.0187}$	$\mathbf{-4.2993}_{\pm 0.0721}$

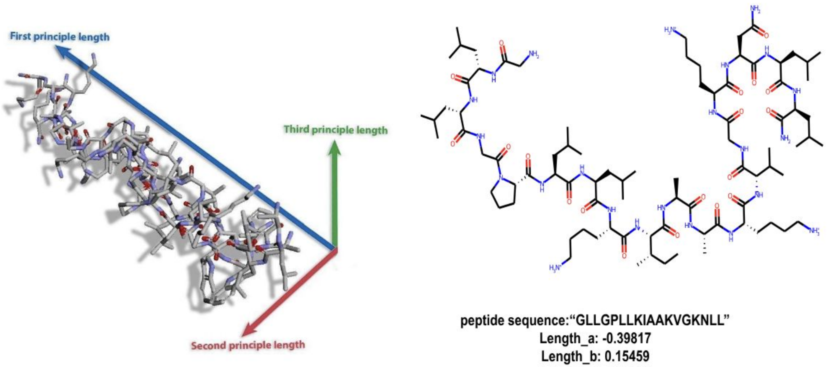
1.98 log(MSE) better *than the best MPNN on average*

0.33 log(MSE) better

0.81 log(MSE) better *than the best baseline on average*

Long Range Graph Benchmark

- Real-world benchmark
  - **Peptide-func** and **Peptide-struct**



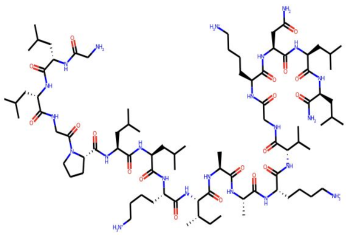
peptide sequence:"GLLGPLLKIAAKVGKNLL"
Length_a: -0.39817
Length_b: 0.15459
Length_c: -0.27048

Long Range Graph Benchmark

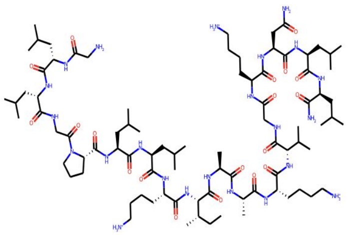
- Real-world benchmark
  - **Peptide-func** and **Peptide-struct**

- Multi-label graph **classif**.
- Predict the peptide **function**

- Multi-dim graph **regression**
- Predict **structural properties** of peptides



Is it an antiviral peptide?
Is it an antibacterial peptide?
...



Inertia mass?
Length?
Sphericity?
...

Long Range Graph Benchmark

Table 2: Results for Peptides-func and Peptides-struct averaged over 3 training seeds. The first, second, and **third** best scores are colored.

Model	Peptides-func AP ↑	Peptides-struct MAE ↓
MPNNs		
GCN	$0.6860_{\pm 0.0050}$	$\mathbf{0.2460}_{\pm 0.0007}$
GCNII	$0.5543_{\pm 0.0078}$	$0.3471_{\pm 0.0010}$
GINE	$0.6621_{\pm 0.0067}$	$0.2473_{\pm 0.0017}$
GatedGCN	$0.6765_{\pm 0.0047}$	$0.2477_{\pm 0.0009}$
Multi-hop DGNs		
DIGL+MPNN+LapPE	$0.6830_{\pm 0.0026}$	$0.2616_{\pm 0.0018}$
MixHop-GCN+LapPE	$0.6843_{\pm 0.0049}$	$0.2614_{\pm 0.0023}$
DRew-GCN+LapPE	$\mathbf{0.7150}_{\pm 0.0044}$	$0.2536_{\pm 0.0015}$
Transformers		
Transformer+LapPE	$0.6326_{\pm 0.0126}$	$0.2529_{\pm 0.0016}$
SAN+LapPE	$0.6384_{\pm 0.0121}$	$0.2683_{\pm 0.0043}$
GraphGPS+LapPE	$0.6535_{\pm 0.0041}$	$\mathbf{0.2500}_{\pm 0.0005}$
DE-DGNs		
GRAND	$0.5789_{\pm 0.0062}$	$0.3418_{\pm 0.0015}$
GraphCON	$0.6022_{\pm 0.0068}$	$0.2778_{\pm 0.0018}$
A-DGN	$0.5975_{\pm 0.0044}$	$0.2874_{\pm 0.0021}$
Ours		
H-DGN	$\mathbf{0.6961}_{\pm 0.0070}$	$0.2581_{\pm 0.0020}$
PH-DGN	$\mathbf{0.7012}_{\pm 0.0045}$	$\mathbf{0.2465}_{\pm 0.0020}$

Long Range Graph Benchmark

include PE and SE

Table 2: Results for Peptides-func and Peptides-struct averaged over 3 training seeds. The **first**, **second**, and **third** best scores are colored.

Model	Peptides-func AP ↑	Peptides-struct MAE ↓
MPNNs		
GCN	$0.6860_{\pm 0.0050}$	$\mathbf{0.2460}_{\pm 0.0007}$
GCNII	$0.5543_{\pm 0.0078}$	$0.3471_{\pm 0.0010}$
GINE	$0.6621_{\pm 0.0067}$	$0.2473_{\pm 0.0017}$
GatedGCN	$0.6765_{\pm 0.0047}$	$0.2477_{\pm 0.0009}$
Multi-hop DGNs		
DIGL+MPNN+LapPE	$0.6830_{\pm 0.0026}$	$0.2616_{\pm 0.0018}$
MixHop-GCN+LapPE	$0.6843_{\pm 0.0049}$	$0.2614_{\pm 0.0023}$
DRew-GCN+LapPE	$\mathbf{0.7150}_{\pm 0.0044}$	$0.2536_{\pm 0.0015}$
Transformers		
Transformer+LapPE	$0.6326_{\pm 0.0126}$	$0.2529_{\pm 0.0016}$
SAN+LapPE	$0.6384_{\pm 0.0121}$	$0.2683_{\pm 0.0043}$
GraphGPS+LapPE	$0.6535_{\pm 0.0041}$	$\mathbf{0.2500}_{\pm 0.0005}$
DE-DGNs		
GRAND	$0.5789_{\pm 0.0062}$	$0.3418_{\pm 0.0015}$
GraphCON	$0.6022_{\pm 0.0068}$	$0.2778_{\pm 0.0018}$
A-DGN	$0.5975_{\pm 0.0044}$	$0.2874_{\pm 0.0021}$
Ours		
H-DGN	$\mathbf{0.6961}_{\pm 0.0070}$	$0.2581_{\pm 0.0020}$
PH-DGN	$\mathbf{0.7012}_{\pm 0.0045}$	$\mathbf{0.2465}_{\pm 0.0020}$

Long Range Graph Benchmark

include PE and SE

Better than MPNN, DE-DGN, graph transformers and rewiring

Table 2: Results for Peptides-func and Peptides-struct averaged over 3 training seeds. The first, second, and **third** best scores are colored.

Model	Peptides-func AP ↑	Peptides-struct MAE ↓
MPNNs		
GCN	$0.6860_{\pm 0.0050}$	$0.2460_{\pm 0.0007}$
GCNII	$0.5543_{\pm 0.0078}$	$0.3471_{\pm 0.0010}$
GINE	$0.6621_{\pm 0.0067}$	$0.2473_{\pm 0.0017}$
GatedGCN	$0.6765_{\pm 0.0047}$	$0.2477_{\pm 0.0009}$
Multi-hop DGNs		
DIGL+MPNN+LapPE	$0.6830_{\pm 0.0026}$	$0.2616_{\pm 0.0018}$
MixHop-GCN+LapPE	$0.6843_{\pm 0.0049}$	$0.2614_{\pm 0.0023}$
DRew-GCN+LapPE	$0.7150_{\pm 0.0044}$	$0.2536_{\pm 0.0015}$
Transformers		
Transformer+LapPE	$0.6326_{\pm 0.0126}$	$0.2529_{\pm 0.0016}$
SAN+LapPE	$0.6384_{\pm 0.0121}$	$0.2683_{\pm 0.0043}$
GraphGPS+LapPE	$0.6535_{\pm 0.0041}$	$\mathbf{0.2500}_{\pm 0.0005}$
DE-DGNs		
GRAND	$0.5789_{\pm 0.0062}$	$0.3418_{\pm 0.0015}$
GraphCON	$0.6022_{\pm 0.0068}$	$0.2778_{\pm 0.0018}$
A-DGN	$0.5975_{\pm 0.0044}$	$0.2874_{\pm 0.0021}$
Ours		
H-DGN	$\mathbf{0.6961}_{\pm 0.0070}$	$0.2581_{\pm 0.0020}$
PH-DGN	$0.7012_{\pm 0.0045}$	$0.2465_{\pm 0.0020}$

Conclusions

- PH-DGN is a new framework for DGNs
  - **Non-dissipative** and **non-conservative** behaviors equilibrium
  - Effectively explore long-range dependencies
- PH-DGN can be used to reinterpret and extend any classical DGN
- Our results show that:
  - PH-DGN outperforms state-of-the-art models
  - Data-driven forces maximize long-range propagation efficacy

References

[1] Di Giovanni, Francesco, et al. “Understanding convolution on graphs via energies”. Transactions on Machine Learning Research (2023).

[2] Desai, Shaan A., et al. “Port-Hamiltonian neural networks for learning explicit time-dependent dynamical systems”. Phys. Rev. E 104 (3 Sept. 2021): 034312.

[3] Galimberti, Clara Lucía, et al. “Hamiltonian Deep Neural Networks Guaranteeing Nonvanishing Gradients by Design”. IEEE Transactions on Automatic Control 68, no. 5 (2023): 3155–3162.

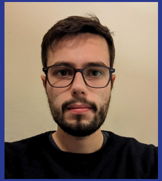
[4] Gravina, Alessio, Davide Bacciu, and Claudio Gallicchio. “Anti-Symmetric DGN: a stable architecture for Deep Graph Networks”. In The Eleventh International Conference on Learning Representations. 2023.

[5] Meyer, Kenneth R, and Daniel C Offin. Introduction to Hamiltonian Dynamical Systems and the N-Body Problem. 3rd ed. Springer, Cham, 2017.

The Team

Simon Heilig

Alessio Gravina



Alessandro Trenta

Claudio Gallicchio

Davide Bacciu

Thank you for your attention

Arxiv

Simon Heilig

simon99.heilig@gmail.com

https://simonheilig.github.io/

Dept. of Computer Science, Ruhr University Bochum, Germany

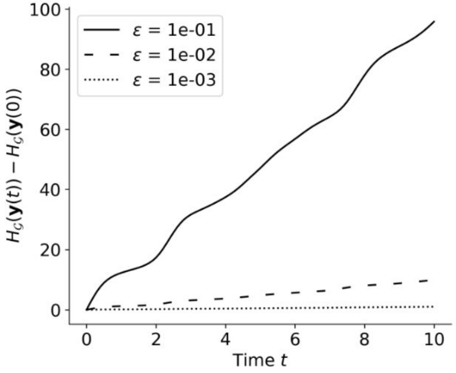
Alessio Gravina

alessio.gravina@phd.unipi.it

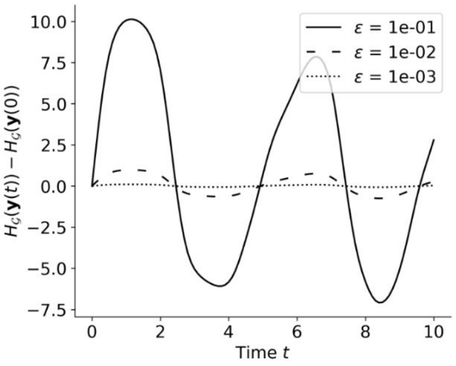
http://pages.di.unipi.it/gravina/

Dept. of Computer Science, University of Pisa, Italy

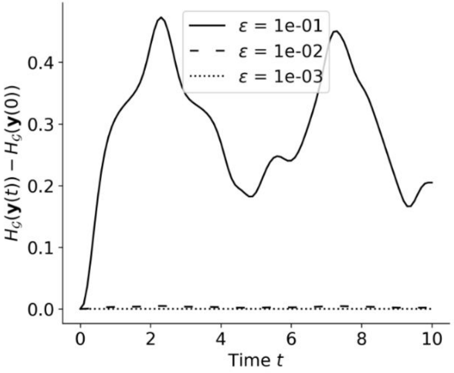
Backup I



Forward Euler



Symplectic Euler



Strömer-Verlet

method	H error	solution error
Forward Euler	$\mathcal{O}(t\epsilon)$	$\mathcal{O}(t^2\epsilon)$
Symplectic Euler	$\mathcal{O}(\epsilon)$	$\mathcal{O}(t\epsilon)$
Störmer-Verlet	$\mathcal{O}(\epsilon^2)$	$\mathcal{O}(t\epsilon^2)$

Backup II

Architecture

Coupling: $\Phi_{\mathbf{p}}, \Phi_{\mathbf{q}} \in \{$Vanilla , GCN$\}$

Decoding: $\{\mathbf{p}, \mathbf{q}, (\mathbf{p}, \mathbf{q})\}$

Dimension: $d \in \{10, 20, 30\}$

Port-Hamiltonian

Dampening: $$D_u(\mathbf{q}) = \begin{cases} \mathbf{w} \in \mathbb{R}^d \\ \mathrm{ReLU}(\mathbf{w}) \in \mathbb{R}^d \\ \mathrm{lin}(\tilde{d}, d) \circ \mathrm{ReLU} \circ \mathrm{lin}(\tilde{d}, \tilde{d}) \circ \mathrm{ReLU} \circ \mathrm{lin}(\tilde{d}, \tilde{d}) \circ \mathrm{ReLU} \circ \mathrm{lin}(d, \tilde{d}) \\ \mathrm{ReLU} \circ \sum_{i \in \mathcal{N}_v} \mathrm{lin}(d, d) \end{cases}$$

External Force: $$F_u(\mathbf{q}, t) = \begin{cases} \mathrm{lin}(\tilde{d}, d) \circ \sin \circ \mathrm{lin}(\tilde{d}, \tilde{d}) \circ \sin \circ \mathrm{lin}(\tilde{d}, \tilde{d}) \circ \sin \circ \mathrm{lin}(d+1, \tilde{d}) \\ \sigma \circ \sum_{i \in \mathcal{N}_v} \mathrm{lin}(d+1, d) \end{cases}$$

Integration

Layers: $L \in \{1, 5, 10, 20, 30\}$

Time: $T \in \{0.1, 1, 2, 3\}$

Backup III

H-DGN:

$$\mathbf{x}_u^{(\ell+1)} = \begin{pmatrix} \mathbf{p}_u^{(\ell+1)} \\ \mathbf{q}_u^{(\ell+1)} \end{pmatrix} = \begin{pmatrix} \mathbf{p}_u^{(\ell)} \\ \mathbf{q}_u^{(\ell)} \end{pmatrix} + \epsilon \mathcal{J}_u \begin{pmatrix} \nabla_{\mathbf{p}_u} H_{\mathcal{G}}(\mathbf{p}^{(\ell+1)}, \mathbf{q}^{(\ell)}) \\ \nabla_{\mathbf{q}_u} H_{\mathcal{G}}(\mathbf{p}^{(\ell+1)}, \mathbf{q}^{(\ell)}) \end{pmatrix}, \quad \forall u \in \mathcal{V}$$

$$\mathbf{p}_u^{(\ell+1)} = \mathbf{p}_u^{(\ell)} - \epsilon \Bigg[\mathbf{W}_q^\top \sigma(\mathbf{W}_q \mathbf{q}_u^{(\ell)} + \Phi_{\mathcal{G}}(\{\mathbf{q}_v^{(\ell)}\}_{v \in \mathcal{N}_u}) + \mathbf{b}_q) + \sum_{v \in \mathcal{N}_u \setminus \{u\}} \mathbf{V}_q^\top \sigma(\mathbf{W}_q \mathbf{q}_v^{(\ell)} + \Phi_{\mathcal{G}}(\{\mathbf{q}_j^{(\ell)}\}_{j \in \mathcal{N}_v}) + \mathbf{b}_q) \Bigg]$$

$$\mathbf{q}_u^{(\ell+1)} = \mathbf{q}_u^{(\ell)} + \epsilon \Bigg[\mathbf{W}_p^\top \sigma(\mathbf{W}_p \mathbf{p}_u^{(\ell+1)} + \Phi_{\mathcal{G}}(\{\mathbf{p}_v^{(\ell+1)}\}_{v \in \mathcal{N}_u}) + \mathbf{b}_p) + \sum_{v \in \mathcal{N}_u \setminus \{u\}} \mathbf{V}_p^\top \sigma(\mathbf{W}_p \mathbf{p}_v^{(\ell+1)} + \Phi_{\mathcal{G}}(\{\mathbf{p}_j^{(\ell+1)}\}_{j \in \mathcal{N}_v}) + \mathbf{b}_p) \Bigg].$$

PH-DGN:

$$\mathbf{p}_u^{(\ell+1)} = \mathbf{p}_u^{(\ell)} + \epsilon \Bigg[-\nabla_{\mathbf{q}_u} H_{\mathcal{G}}(\mathbf{p}^{(\ell)}, \mathbf{q}^{(\ell+1)}) - D_u(\mathbf{q}^{(\ell+1)}) \nabla_{\mathbf{p}_u} H_{\mathcal{G}}(\mathbf{p}^{(\ell)}, \mathbf{q}^{(\ell+1)}) + F_u(\mathbf{q}^{(\ell+1)}, t) \Bigg].$$