GRAND: Graph Neural Diffusion

Benjamin P. Chamberlain, James Rowbottom, Maria Gorinova, Stefan Webb, Emanuele Rossi, Michael M. Bronstein.

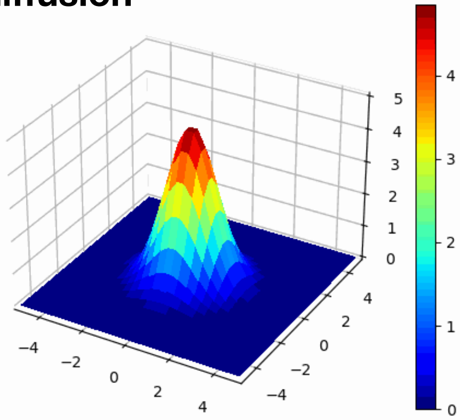
GRAND: Graph Neural Diffusion

Linking

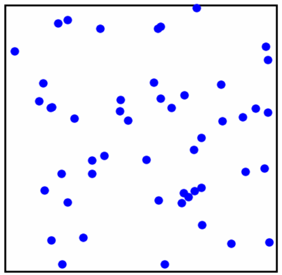
- diffusion processes
- GNNs
- partial differential equations
  - leading to theoretical foundations and new architectures for GNNs

Diffusion PDEs arise in many physical processes involving the transfer of "stuff"

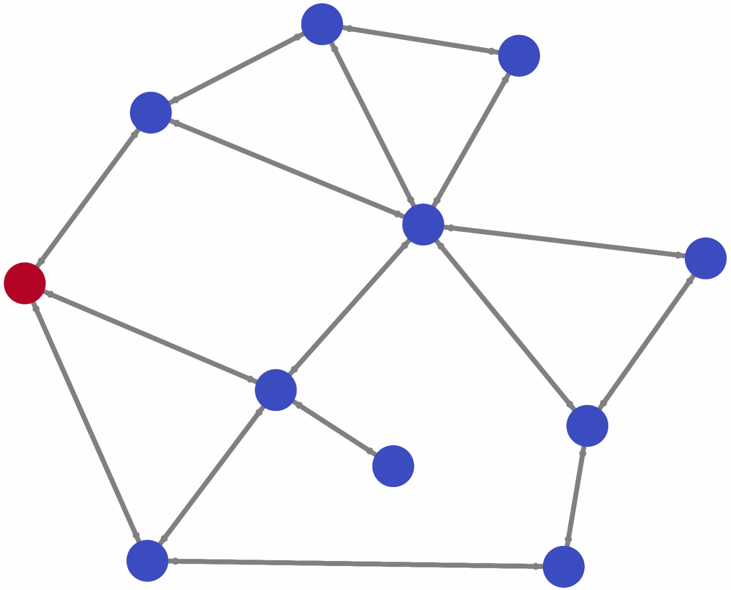
Heat diffusion



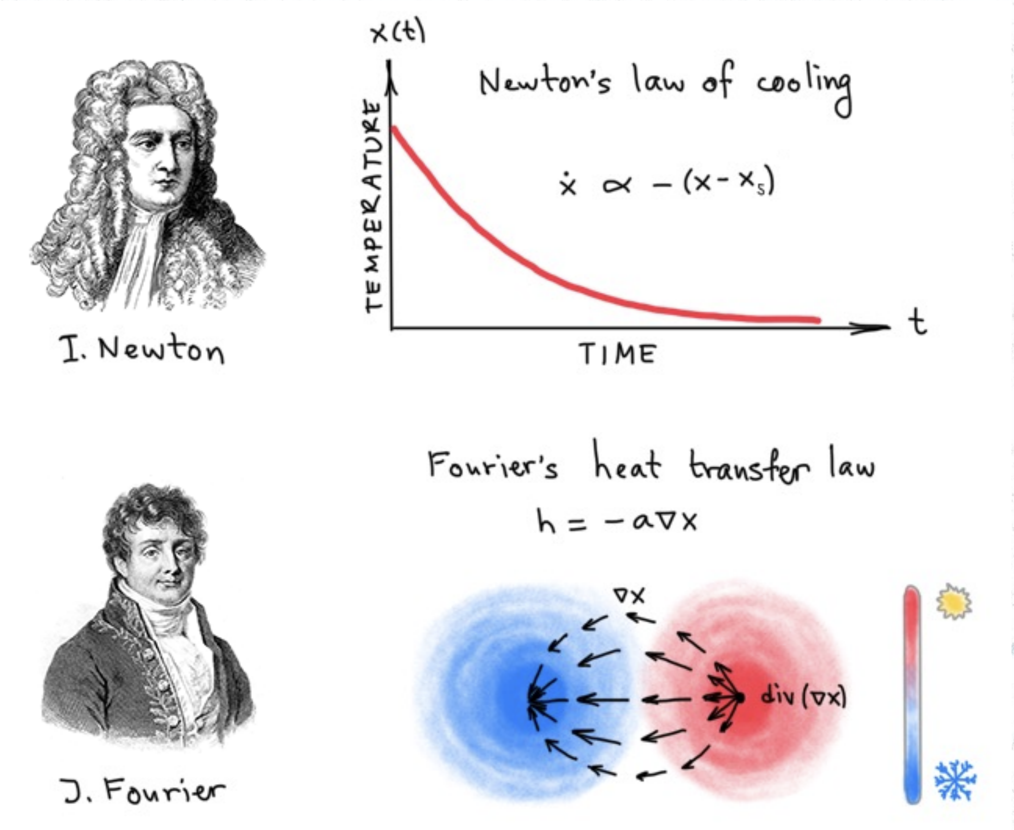
Particle diffusion



Laplacian diffusion on graphs

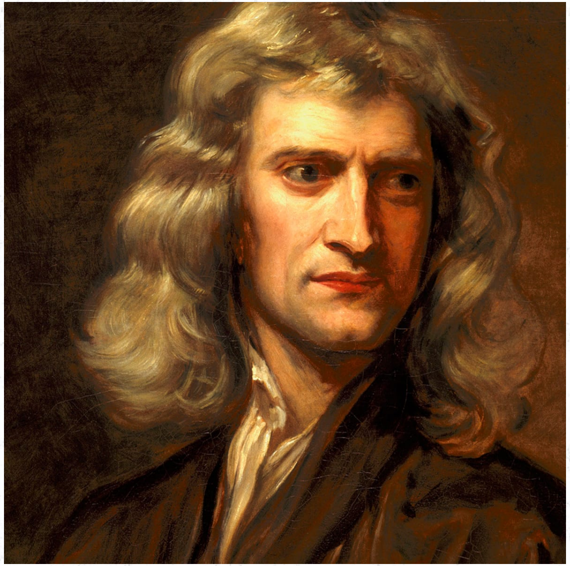


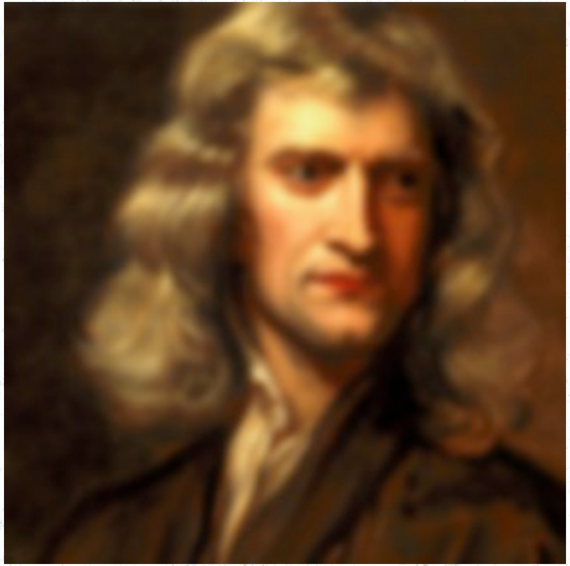
A diffusion process can be expressed as a partial differential equation



$$\frac{dx}{dt} = \mathrm{div}(a(u,t)\nabla x(u,t))$$

Diffusion in Image processing





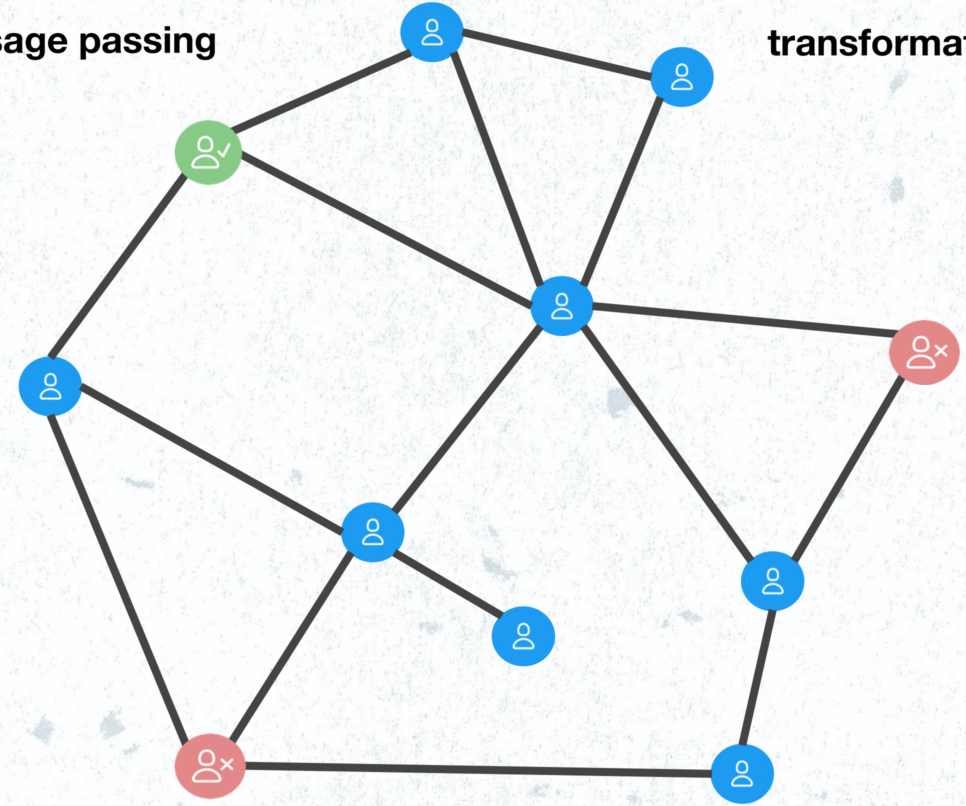
Message passing neural network as diffusion of information on a graph

input

message passing

transformation

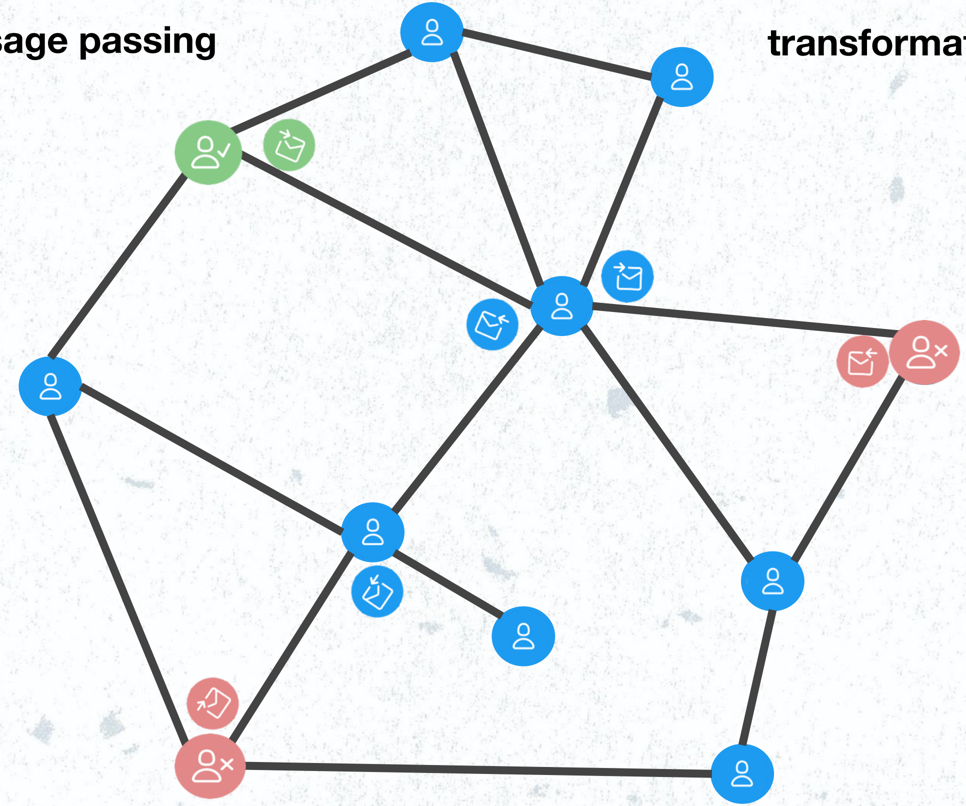
output



Message passing neural network as diffusion of information on a graph

input

message passing



transformation

output

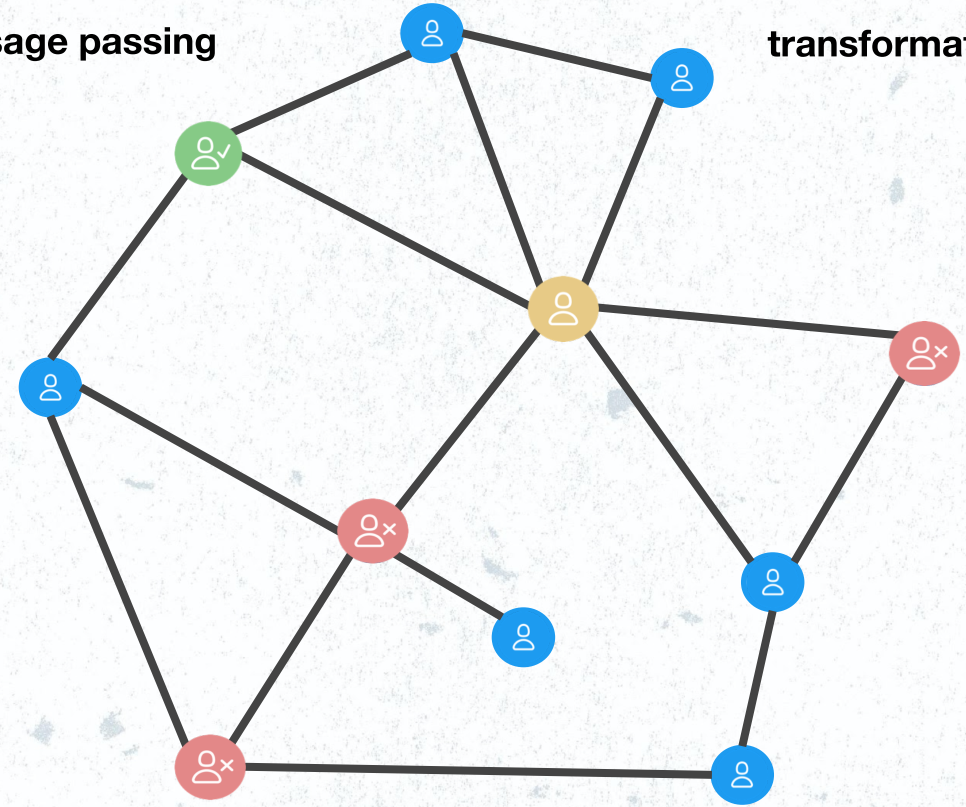
Message passing neural network as diffusion of information on a graph

input

message passing

transformation

output



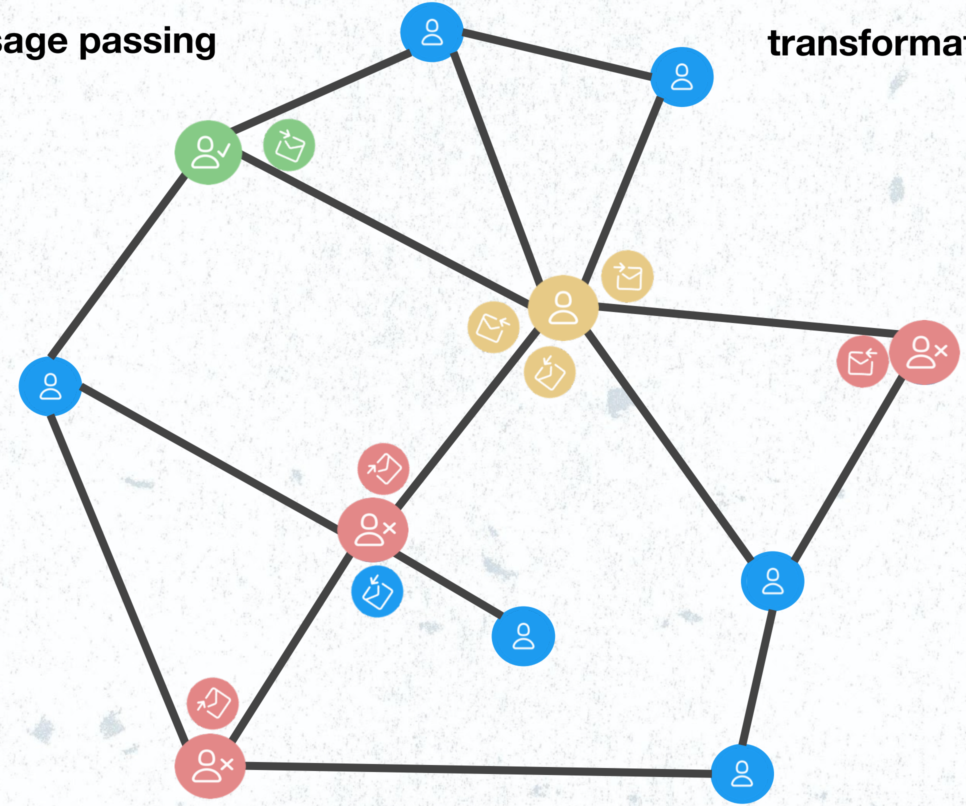
Message passing neural network as diffusion of information on a graph

input

message passing

transformation

output



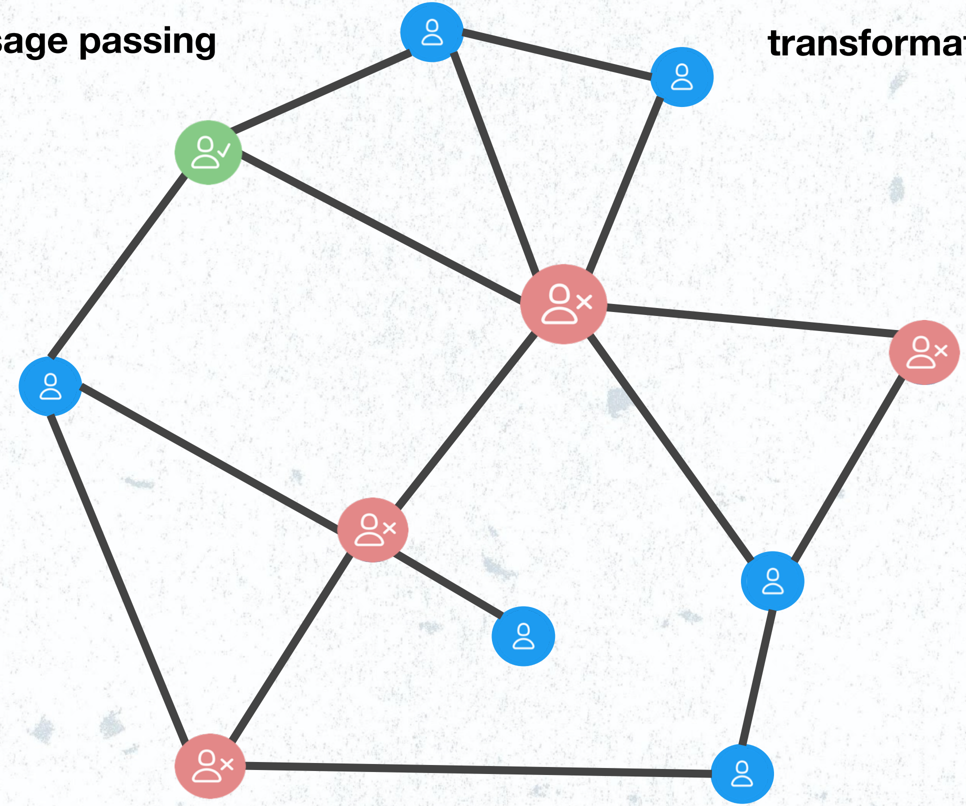
Message passing neural network as diffusion of information on a graph

input

message passing

transformation

output



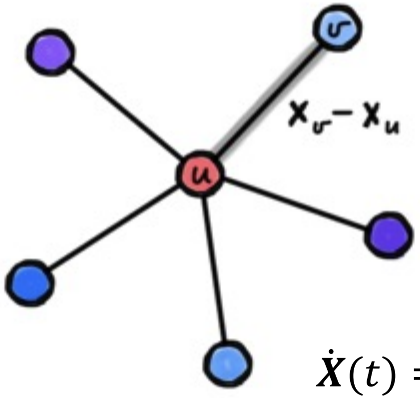
Gradient, divergence and diffusion on graphs

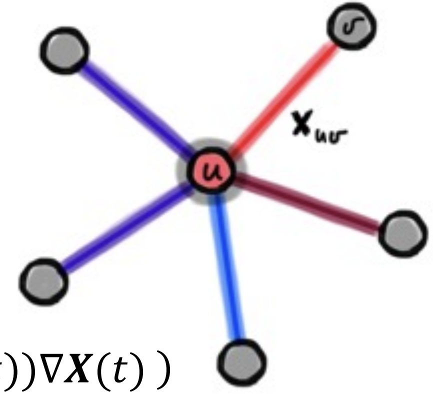
gradient - flow along edges

$$(\nabla \boldsymbol{X})_{uv} = \boldsymbol{x}_u - \boldsymbol{x}_v$$

divergence - aggregation of edges

$$\big(\mathrm{div}(\boldsymbol{X})\big)_u = \sum_v w_{uv} x_{uv}$$



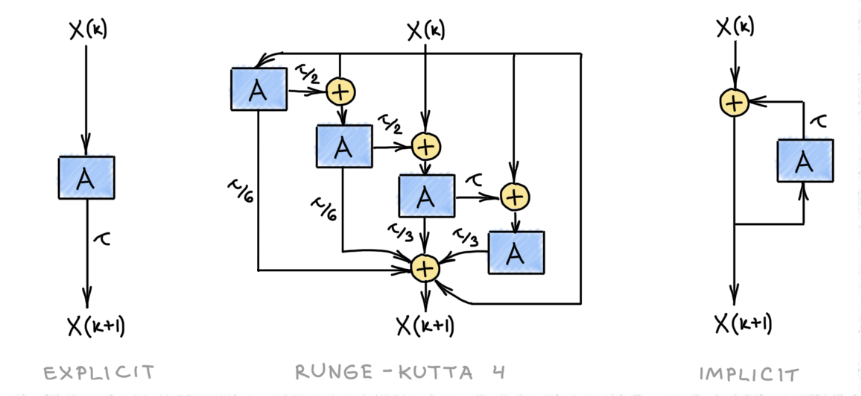


$$\dot{\boldsymbol{X}}(t) = \mathrm{div}(\boldsymbol{A}(\boldsymbol{X}(t))\nabla \boldsymbol{X}(t)\,)$$

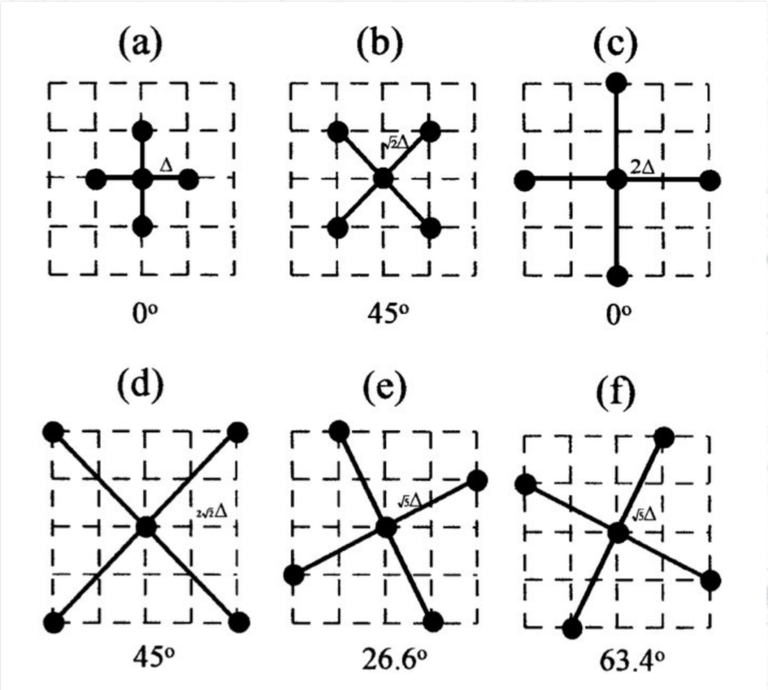
$$\frac{[\boldsymbol{X}(k+1) - \boldsymbol{X}(k)]}{\tau} = \big[\boldsymbol{A}\big(\boldsymbol{X}(k)\big) - \boldsymbol{I}\big]\boldsymbol{X}(k)$$

Designing new GNNs

Multi-step and adaptive solvers

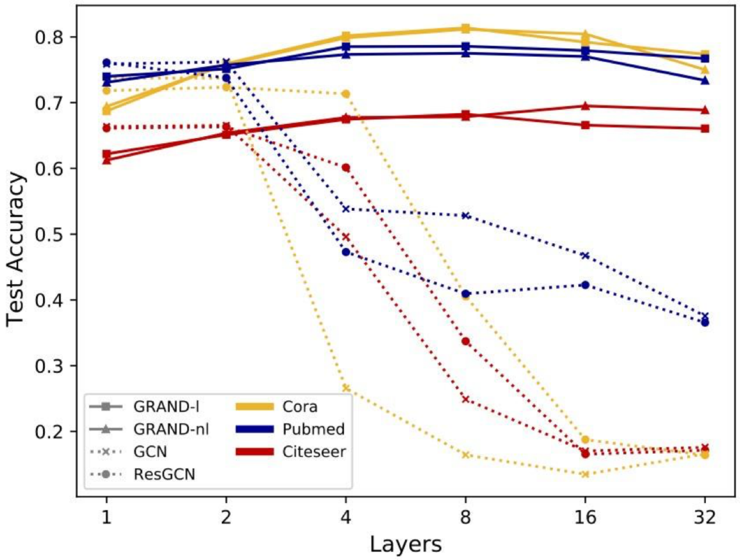


Implicit solvers – multi-hop filters



Random splits	CORA	CiteSeer	PubMed	Coathor CS	Computer	Photo	ogb-arxiv*
GCN	81.5 ± 1.3	**71.9 ± 1.9**	77.8 ± 2.9	91.1 ± 0.5	82.6 ± 2.4	91.2 ± 1.2	**72.17 ± 0.33**
GAT	81.8 ± 1.3	71.4 ± 1.9	**78.7 ± 2.3**	90.5 ± 0.6	78.0 ± 19.0	85.7 ± 20.3	**73.65 ± 0.11**†
GAT-ppr	81.6 ± 0.3	68.5 ± 0.2	76.7 ± 0.3	91.3 ± 0.1	**85.4 ± 0.3**	90.9 ± 0.3	N/A
MoNet	81.3 ± 1.3	71.2 ± 2.0	**78.6 ± 2.3**	90.8 ± 0.6	83.5 ± 2.2	91.2 ± 2.3	N/A
GS-mean	79.2 ± 7.7	71.6 ± 1.9	77.4 ± 2.2	91.3 ± 2.8	82.4 ± 1.8	91.4 ± 1.3	71.39 ± 0.16
GS-maxpool	76.6 ± 1.9	67.5 ± 2.3	76.1 ± 2.3	85.0 ± 1.1	N/A	90.4 ± 1.3	N/A
CGNN	81.4 ± 1.6	66.9 ± 1.8	66.6 ± 4.4	**92.3 ± 0.2**	80.29 ±2.0	91.39 ± 1.5	58.70 ± 2.5
GDE	78.7 ± 2.2	71.8 ± 1.1	73.9 ± 3.7	91.6 ± 0.1	82.9 ± 0.6	**92.4 ± 2.0**	56.66 ± 10.9
GRAND-l (ours)	**83.6 ± 1.0**	**73.4 ± 0.5**	**78.8 ± 1.7**	**92.9 ± 0.4**	**83.7** ± **1.2**	**92.3** ± **0.9**	71.87 ± 0.17
GRAND-nl (ours)	**82.3** ± **1.6**	70.9 ± 1.0	77.5 ± 1.8	**92.4 ± 0.3**	82.4 ± 2.1	**92.4 ± 0.8**	71.2 ± 0.2
GRAND-nl-rw (ours)	**83.3 ± 1.3**	**74.1 ± 1.7**	78.1 ± 2.1	91.3 ± 0.7	**85.8 ± 1.5**	**92.5 ± 1.0**	**72.23** ± **0.20**

Table 2. Test accuracy and std for 20 random initializations and 100 random train-val-test splits. *Using labels. †using 1.5M parameters.



Graph Neural Diffusion provides a principled mathematical framework for
studying many popular architectures for deep learning on graphs as well as a
blueprint for developing new ones

Thank You