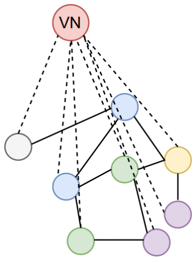
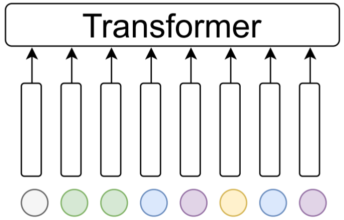
UC San Diego

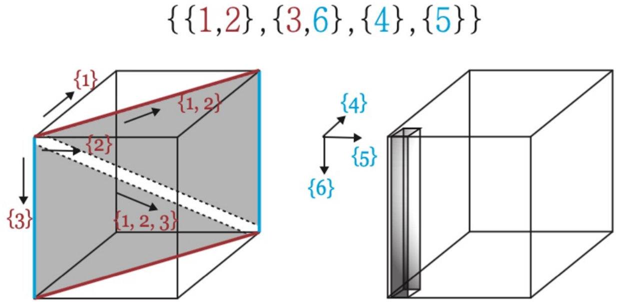
Local-to-Global Perspectives on Graph Networks

Chen Cai

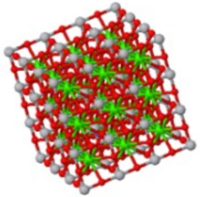


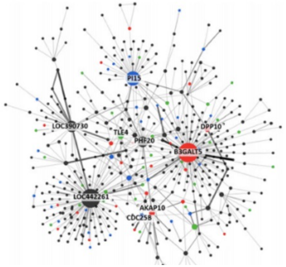


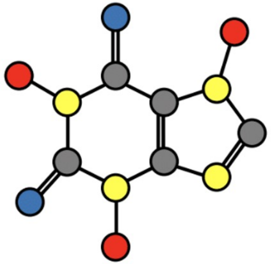
Committee members:
Jingbo Shang
Yusu Wang
Gal Mishne
Rose Yu

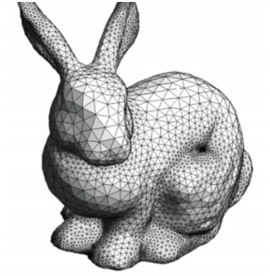


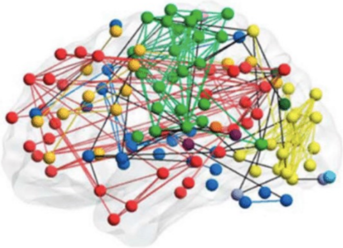
Graphs are everywhere





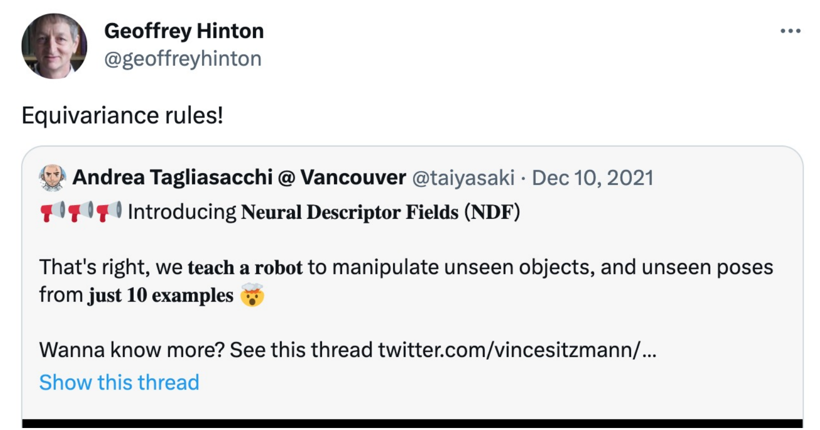


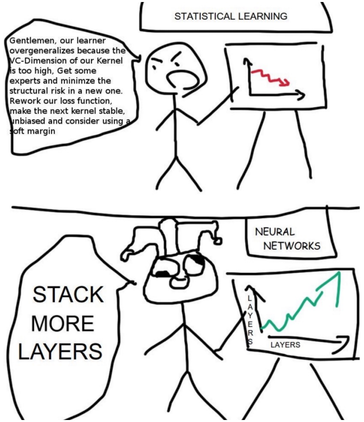




Graph Neural Network

- Generalize CNN to graphs
- Permutation equivariant/invariant $f(PX) = Pf(X) / f(PX) = f(X)$
- Handles rich node/edge scalar/vector/high-order tensor features
- Train on small graphs, generalize to large graphs





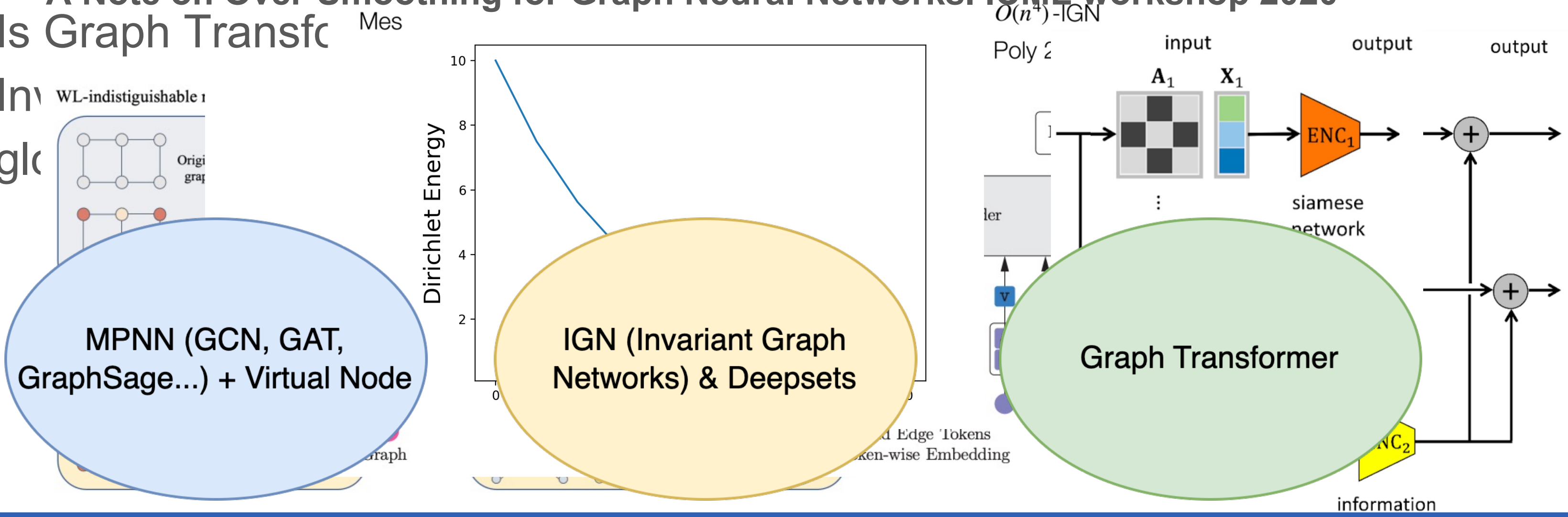
Local vs. Global GNN

- Message Passing Neural Network (MPNN) mix features locally
  - GIN, GCN, GraphSage, GAT….
  - over-squashing, over-smoothing, limited expressive power
- To go from 1-WL to higher WL, one needs to go high-order/global

Equivariant Subgraph Aggregation Networks. ICLR 2022 Spotlight

A Note on Over-Smoothing for Graph Neural Networks. ICML workshop 2020

- Is Graph Transfo
- In
  gl



My research in GNN

Theory

- **Expressive power** of GNN ICLR 2022
- **Over-smoothing** for GNN ICML 2020 workshop

theory of local GNN

- **Convergence** of IGN ICML 2022
- **Connection** between MPNN and GT ICML 2023

theory of global GNN

Application

- **Graph Coarsening** with neural networks ICLR 2021
- **Generative coarse-graining** of molecular conformations ICML 2022

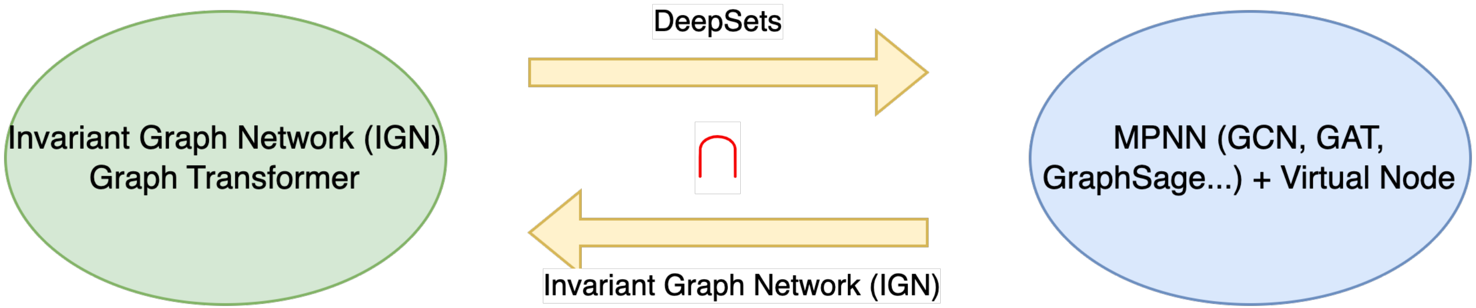
coarsening & de-coarsening

- DeepSets for high-entropy alloys npj Computational Materials
- SO(3) equivariant network for tensor regression IJMCE

property prediction

Agenda

- Intro & research overview (10 min) ✅
- Convergence of Invariant Graph Network ICML 2022 (18 min)
- On the connection between MPNN and Graph Transformer (15 min) ICML 2023
- Graph coarsening with neural networks ICLR 2021 (5 min)
- Conclusion (3 min)

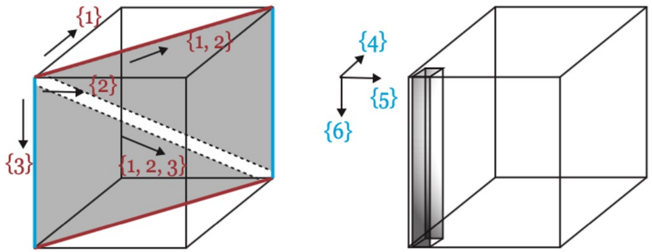


Convergence of Invariant Graph Networks

Chen Cai & Yusu Wang

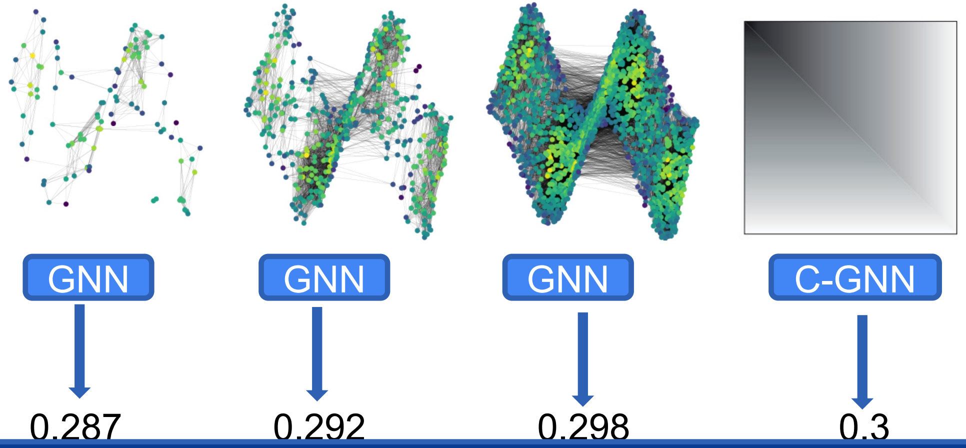
ICML 2022

$\{\{1,2\},\{3,6\},\{4\},\{5\}\}$



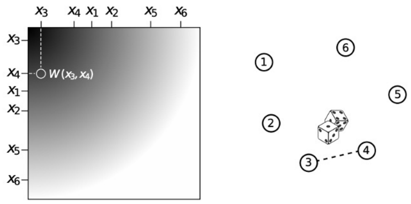
Motivation

- What is convergence?
  - A sequence of graphs are sampled from the same model
  - Send each graph to the same GNN
  - Does output (a sequence of vectors) converge?
- Convergence is easier to tackle than generalization
  - Variability is controlled & limited

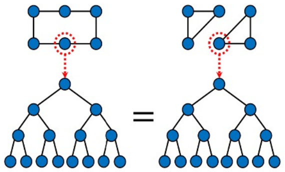


Setup & existing work

- Model
  - graphon $W: [0,1]^2 \to [0,1]$
  - edge probability discrete model
  - edge weight continuous model
- Previous work studied spectral GNN, which has limited expressive power
- What about more powerful GNN?



Connect nodes 3 and 4 with probability $W(x_3, x_4)$.



Study the Convergence of Invariant Graph Networks (IGN)

Keriven et al. "Convergence and stability of graph convolutional networks on large random graphs." NeurIPS 2020.
Ruiz et al. "Graphon neural networks and the transferability of graph neural networks." NeurIPS 2020

Invariant Graph Network (IGN)

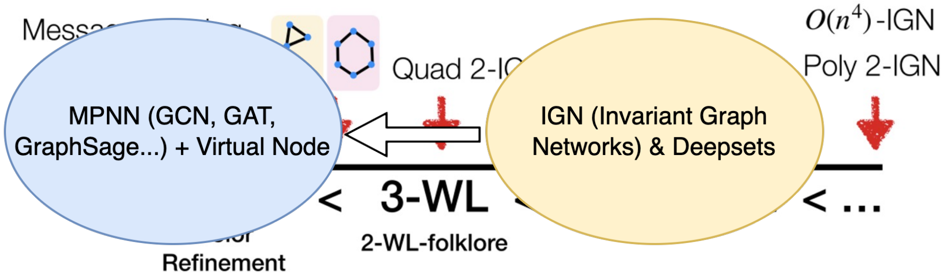
- $F = h \circ L^{(T)} \circ \sigma \cdots \circ \sigma \circ L^{(1)}$ needs to be permutation equivaraint
- Characterize *linear permutation equivariant* functions
- 15 basis elements for $\mathbb{R}^{n^2} \rightarrow \mathbb{R}^{n^2}$
- Generalization of DeepSets

Theorem [Maron et al 2018]: The space of linear permutation equivariant functions $\mathbb{R}^{n^l} \rightarrow \mathbb{R}^{n^m}$ is of dimension $bell(l+m)$, number of partitions of set $\{1, 2, \ldots, l+m\}$.

Maron, Haggai, et al. "Invariant and equivariant graph networks." ICLR 2019
Zaheer, Manzil, et al. "Deep sets." NeurIPS 2017

Invariant Graph Network (IGN)

- Depending on largest intermediate tensor order, we have 2-IGN and k-IGN
- 2-IGN:
  - Can approximate Message Passing neural network (MPNN)
  - At least as powerful as 1-WL (Weisfeiler-Leman Algorithm)
- k-IGN
  - Not practical but a good mental model for GNN expressivity research
  - As k increase, k-IGN reaches universality



Maron, Haggai, et al. "Invariant and equivariant graph networks." ICLR 2019
Maron, Haggai, et al. "Provably powerful graph networks. NeurIPS 2019

Convergence for 2-IGN

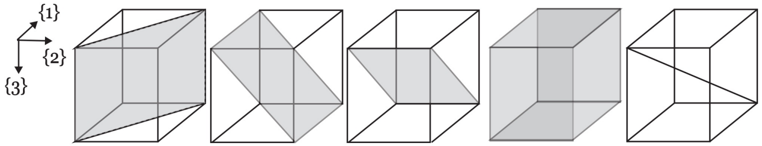
- Analysis of basis elements one by one
- Spectral norm of some elements is unbounded
- Introducing "partition norm"

Definition (**Partition-norm**): The partition-norm of 2-tensor $A \in \mathbb{R}^{n^2}$ is defined as $||A||_{pn} := \left(\frac{Diag^*(A)}{\sqrt{n}}, \frac{||A||_2}{n}\right)$. The continuous analog of the partition-norm for graphon $W \in \mathcal{W}$ is defined as $||W||_{pn} := \left(\sqrt{\int W^2(u,u)du}, \sqrt{\int W^2(u,v)dudv}\right)$

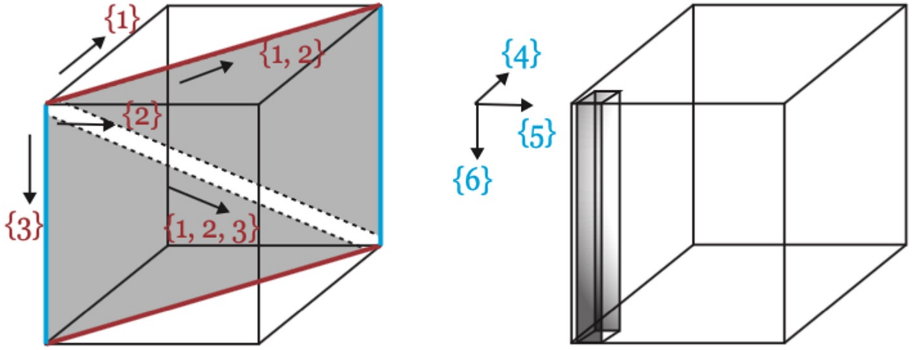
$$\forall i \in [15], if\, ||A||_{pn} \leq (\epsilon, \epsilon), then\, ||T_i(A)||_{pn} \leq (\epsilon, \epsilon)$$

Space of linear permutation equivariant maps

- from l-tensor to m-tensor
- dimension is $bell(l+m)$

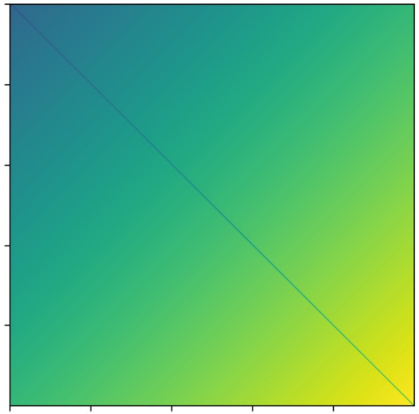


$$\{\{1,2\},\{3,6\},\{4\},\{5\}\}$$

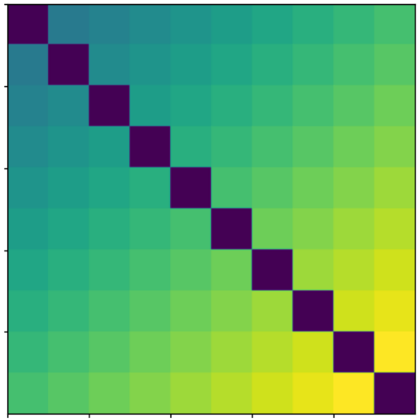


$$\underbrace{S_1=\{\{1,2\}\}}_{\text{Only has input axis}}\cup\underbrace{S_2=\{\{3,6\}\}}_{\text{has both input and output axis}}\cup\underbrace{S_3=\{\{4\},\{5\}\}}_{\text{only has output axis}}$$

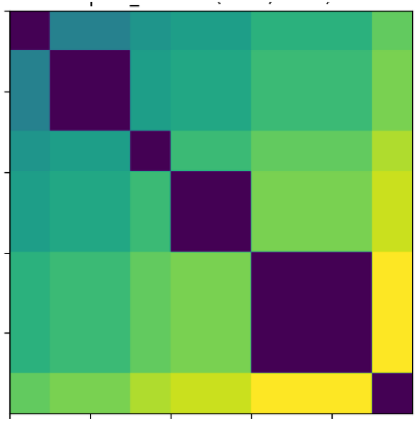
Edge Weight Continuous Model



W



W_n



$\widetilde{W}_n$

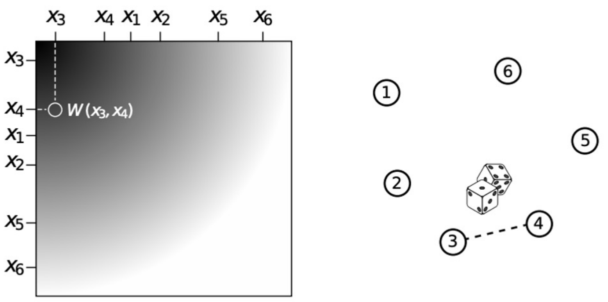
$cIGN(W_n) \to cIGN(W)$

$cIGN(\widetilde{W}_n) \to cIGN(W)$ in probability

Edge Probability Discrete Model

$$RMSE_U(\phi_c(W), \phi_d(A_n))$$

- U is the sampling data
- S_U is the sampling operator
- Comparison in the discrete space
- More natural and more challenging



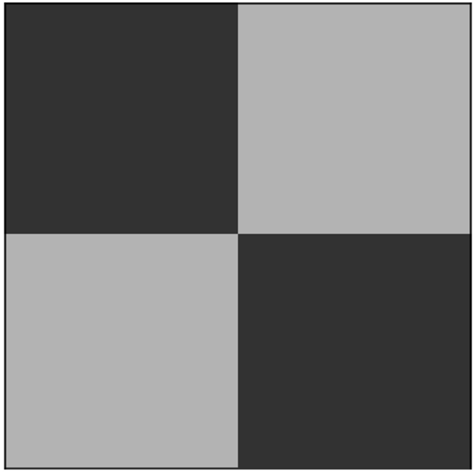
Connect nodes 3 and 4 with probability $W(x_3, x_4)$.

$$RMSE_U(f,x) := \left\| S_u f - \frac{x}{n} \right\|_2 = \left(n^{-2} \sum_{i=0}^{n} \sum_{j=0}^{n} \left\| f(u_i, u_j) - x(i,j) \right\|^2 \right)^{1/2}$$

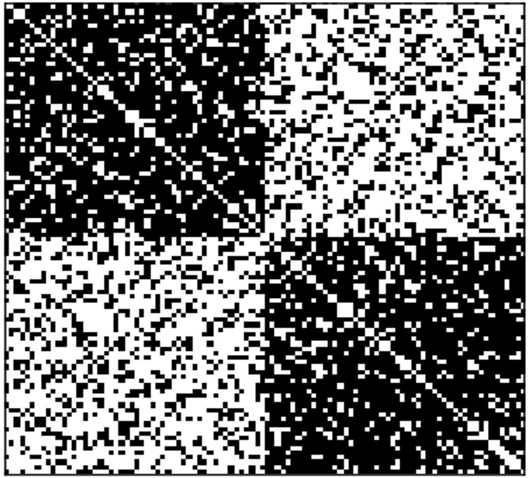
Negative result

Informal Theorem (**negative result**) [Cai & Wang, 2022]
Under mild assumption on W, given any IGN architecture, there exists a set of parameter θ such that the convergence of IGN to cIGN is not possible, i.e., $RMSE_U(\phi_c([W, Diag(X)]), \phi_d([A_n, Diag(\widetilde{X_n})]))$ does not converge to 0 as n goes to infinity, where A_n is 0-1 matrix.

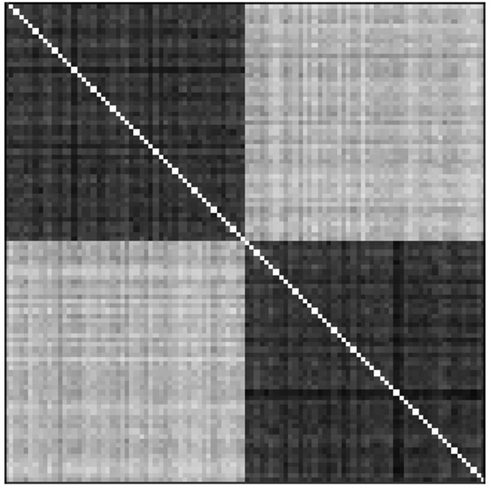
Graphon (edge probability) estimation



W



A_n



$\widehat{W_{n \times n}}$

Does $RMSE_U(\Phi_c(W), \Phi_d(\widehat{W_{n \times n}}))$converges to 0 in probability?

Convergence after edge smoothing

Informal Theorem (**convergence of IGN-small**) [Cai & Wang, 2022]
Assume AS 1-4, and let $\widehat{W_{n\times n}}$ be the estimated edge probability that satisfies $\frac{1}{n}\left\|W_{n\times n} - \widehat{W_{n\times n}}\right\|_2$ converges to 0 in probability. Let Φ_c, Φ_d be continuous and discrete IGN-small. Then $RMSE_U(\phi_c([W, Diag(X)]), \phi_d([\widehat{W_{n\times n}}, Diag(\widetilde{X_n})])$ converges to 0 in probability.

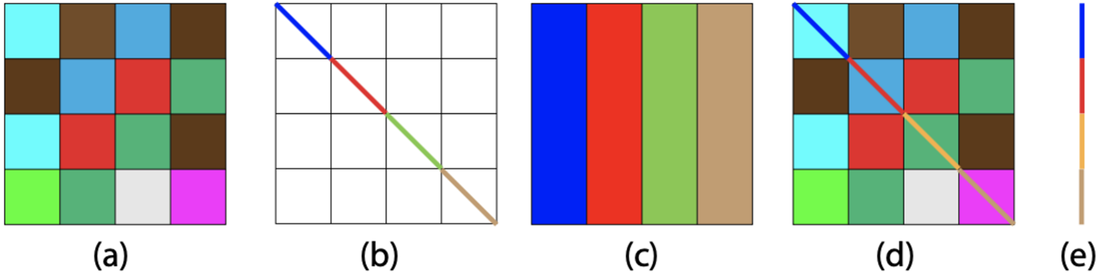
- Proof leverages
  - Statistical guarantee of edge smoothing
  - Property of basis elements of k-IGN
  - Standard algebraic manipulation
  - Property of sampling operator

$$
\begin{aligned}
&\mathrm{RMSE}_U(\Phi_c(W), \Phi_d(\widehat{W}_{n\times n})) \\
&= \|S_U\Phi_c(W) - \frac{1}{\sqrt{n}}\Phi_d(\widehat{W}_{n\times n})\| \\
&\leqslant \underbrace{\|S_U\Phi_c(W) - S_U\Phi_c(\widetilde{W}_n)\|}_{\text{First term: discrization error}} + \underbrace{\|S_U\Phi_c(\widetilde{W}_n) - \Phi_d S_U(\widetilde{W}_n)\|}_{\text{Second term: sampling error}} \\
&+ \underbrace{\|\Phi_d S_U(\widetilde{W}_n) - \frac{1}{\sqrt{n}}\Phi_d(\widehat{W}_{n\times n})\|}_{\text{Third term: estimation error}}
\end{aligned}
$$

IGN-small

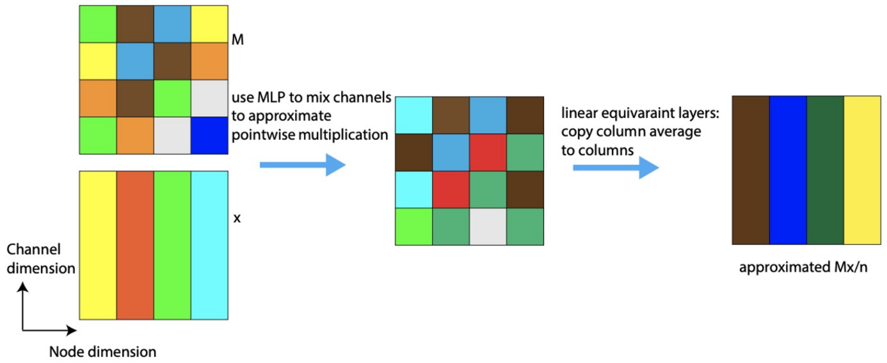
- A subset of IGN

Definition (**IGN-small**): Let $\widetilde{W_{n,E}}$ be a graphon with ``chessboard pattern'', i.e., it is a piecewise constant graphon where each block is of the same size. Similarly, define $\widetilde{X_{n,E}}$ as the 1D analog. IGN-small denotes a subset of IGN that satisfies $S_n \phi_c([\widetilde{W_{n,E}}, Diag(\widetilde{X_{n,E}})]) = \phi_d S_n([\widetilde{W_{n,E}}, Diag(\widetilde{X_{n,E}})])$

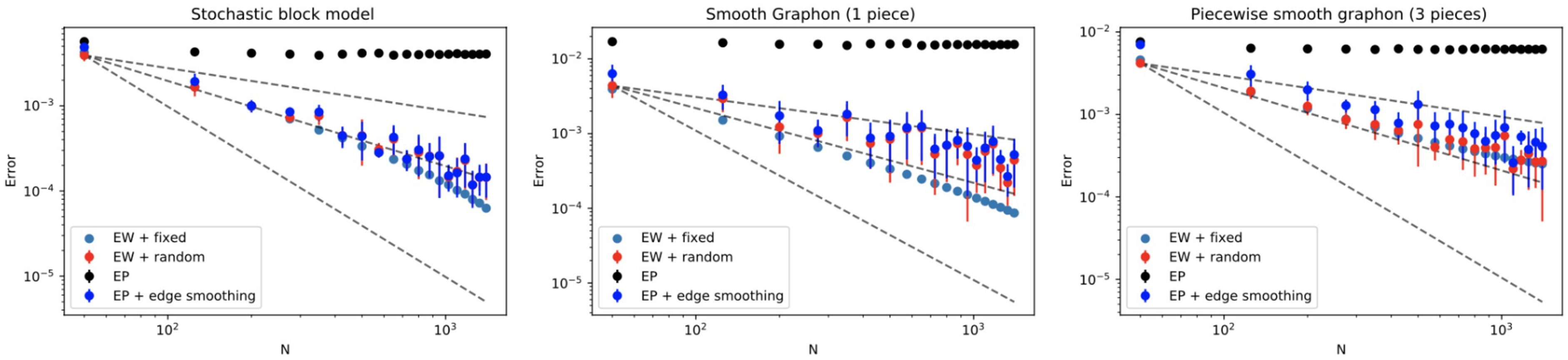


IGN-small can approximate SGNN arbitrarily well

- Spectral GNN (SGNN) $z_j^{(l+1)} = \rho(\sum_{i=1}^{d_l} h_{ij}^{(l)}(L)z_i^{(l)} + b_j^{(l)}1_n)$
- Main GNN considered in the convergence literature
- Proof idea:
    - It suffices to approximate Lx
    - 2-IGN basis elements can compute L and do matrix-vector multiplication



Experiments



Summary

A novel interpretation of basis of the space of equivariant maps in k-IGN

Edge weight *continuous* model:

- Convergence of 2-IGN and k-IGN
- For both deterministic and random sampling

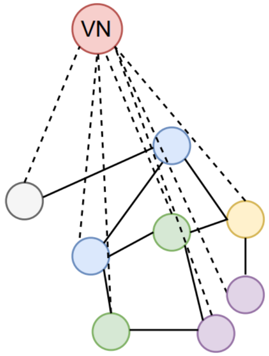
Edge probability *discrete* model

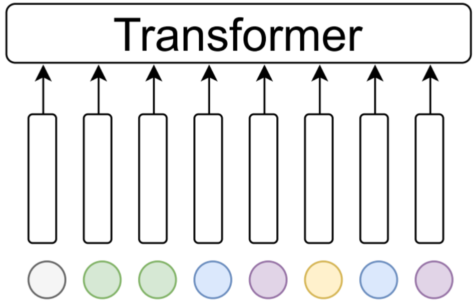
- Negative result in general
- Convergence of IGN-small after graphon estimation
- IGN-small approximates spectral GNN arbitrarily well

On the Connection Between MPNN and Graph Transformer

Chen Cai, Truong Son Hy, Rose Yu, Yusu Wang

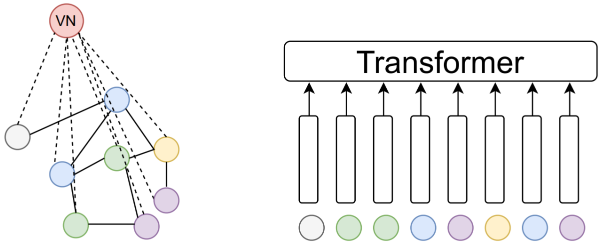
ICML 2023





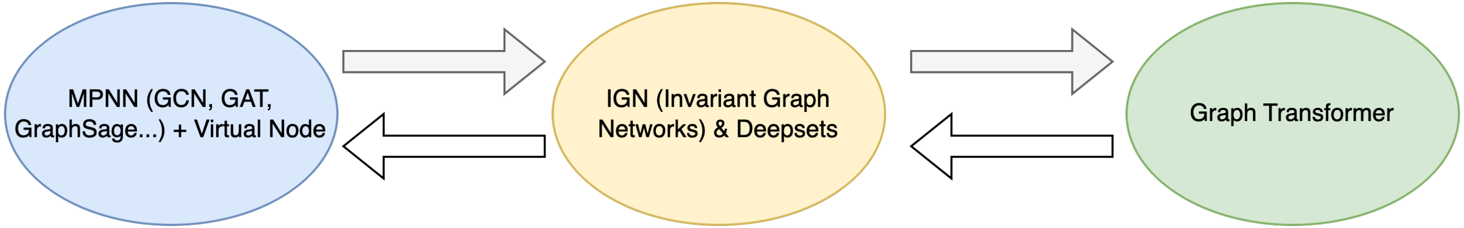
Background

- MPNN: Mixing node features locally
  - GCN, GAT, GIN….
  - Limited expressive power, over-squashing, over-smoothing….
  - Local approach
- GT: tokenize nodes and feed into Transformer
  - Simple; gaining attraction recently
  - Relies on efficient transformer literature to scale up GT
  - Global approach
- What's the *connection* between such two paradigms?



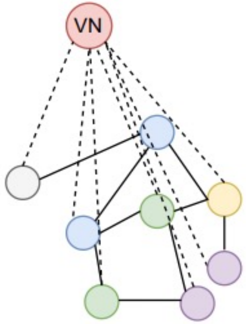
Motivation

- Long range modeling
  - Congestion prediction in chip design, large molecules…
  - Shortcuts, coarsening, graph transformer
- Pure Transformers are powerful graph learners
  - GT with specific positional embedding can approximate 2-IGN, which is at least as expressive as MPNN
  - Proof idea: show that GT can approximate all permutation equivariant layers in IGN
- This paper: draw the inverse connection
  - Can we approximate GT with MPNN?



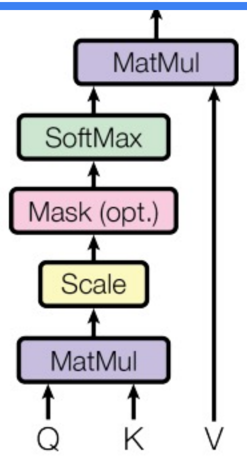
MPNN + Virtual Node (VN)

- Add a virtual node + heterogeneous message passing
- Trivially reduce the diameter to 2
- Proposed in the early days of GNN; commonly used in practice and improves over MPNN
- Very little theoretical understanding
- This paper: show simple MPNN + VN can approximate GT under various width/depth settings



Transformer

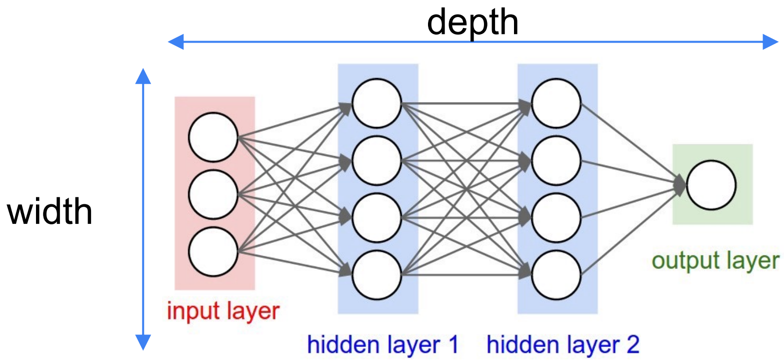
- A sequence of Self-Attention layer
- $L(X) = softmax\left(XW_Q(XW_K)^T\right)XW_V$
- $O(n^2)$ complexity
- Vast literature on efficient/linear transformers
- Behind the success of AF2, LLM, StableDiffusion…



Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

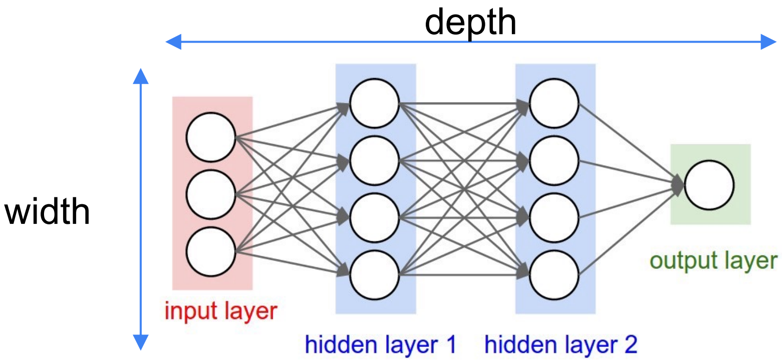
Summary of theoretical results

	Depth	Width	Self-Attention	Note
Theorem 4.1	$\mathcal{O}(1)$	$\mathcal{O}(1)$	Approximate	Approximate self attention in Performer (Choromanski et al., 2020)
Theorem 5.5	$\mathcal{O}(1)$	$\mathcal{O}(n^d)$	Full	Leverage the universality of equivariant DeepSets
Theorem 6.3	$\mathcal{O}(n)$	$\mathcal{O}(1)$	Full	Explicit construction, strong assumption on $\mathcal{X}$
Proposition B.10	$\mathcal{O}(n)$	$\mathcal{O}(1)$	Full	Explicit construction, more relaxed (but still strong) assumption on $\mathcal{X}$



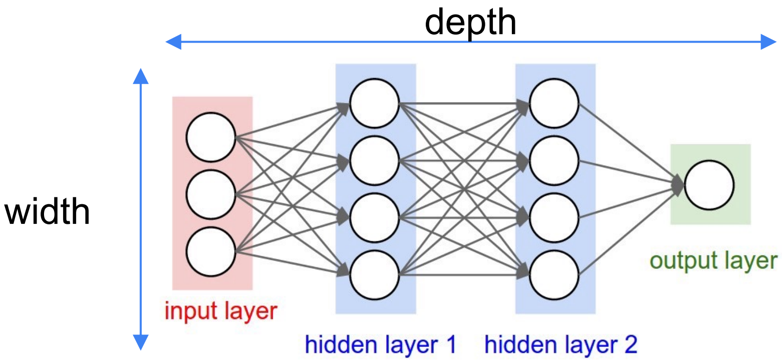
Summary of theoretical results

	Depth	Width	Self-Attention	Note
Theorem 4.1	$\mathcal{O}(1)$	$\mathcal{O}(1)$	Approximate	Approximate self attention in Performer (Choromanski et al., 2020)
Theorem 5.5	$\mathcal{O}(1)$	$\mathcal{O}(n^d)$	Full	Leverage the universality of equivariant DeepSets
Theorem 6.3	$\mathcal{O}(n)$	$\mathcal{O}(1)$	Full	Explicit construction, strong assumption on $\mathcal{X}$
Proposition B.10	$\mathcal{O}(n)$	$\mathcal{O}(1)$	Full	Explicit construction, more relaxed (but still strong) assumption on $\mathcal{X}$



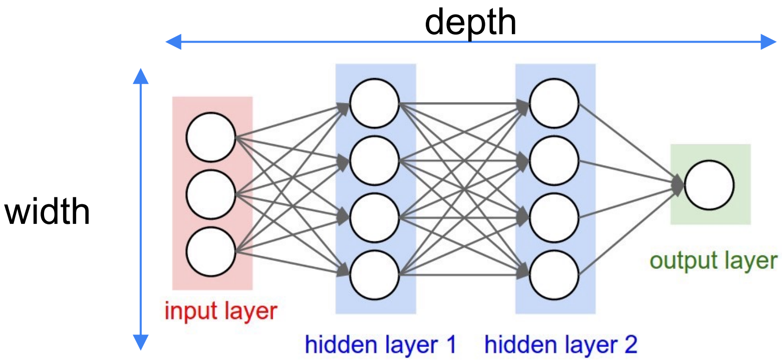
Summary of theoretical results

	Depth	Width	Self-Attention	Note
Theorem 4.1	$\mathcal{O}(1)$	$\mathcal{O}(1)$	Approximate	Approximate self attention in Performer (Choromanski et al., 2020)
Theorem 5.5	$\mathcal{O}(1)$	$\mathcal{O}(n^d)$	Full	Leverage the universality of equivariant DeepSets
Theorem 6.3	$\mathcal{O}(n)$	$\mathcal{O}(1)$	Full	Explicit construction, strong assumption on $\mathcal{X}$
Proposition B.10	$\mathcal{O}(n)$	$\mathcal{O}(1)$	Full	Explicit construction, more relaxed (but still strong) assumption on $\mathcal{X}$



Summary of theoretical results

	Depth	Width	Self-Attention	Note
Theorem 4.1	$\mathcal{O}(1)$	$\mathcal{O}(1)$	Approximate	Approximate self attention in Performer (Choromanski et al., 2020)
Theorem 5.5	$\mathcal{O}(1)$	$\mathcal{O}(n^d)$	Full	Leverage the universality of equivariant DeepSets
Theorem 6.3	$\mathcal{O}(n)$	$\mathcal{O}(1)$	Full	Explicit construction, strong assumption on $\mathcal{X}$
Proposition B.10	$\mathcal{O}(n)$	$\mathcal{O}(1)$	Full	Explicit construction, more relaxed (but still strong) assumption on $\mathcal{X}$



MPNN + VN w/ constant width & depth

- Recall SA layer has the following form

$$\boldsymbol{x}_i^{(l+1)} = \sum_{j=1}^{n} \frac{\kappa\left(\boldsymbol{W}_Q^{(l)}\boldsymbol{x}_i^{(l)}, \boldsymbol{W}_K^{(l)}\boldsymbol{x}_j^{(l)}\right)}{\sum_{k=1}^{n}\kappa\left(\boldsymbol{W}_Q^{(l)}\boldsymbol{x}_i^{(l)}, \boldsymbol{W}_K^{(l)}\boldsymbol{x}_k^{(l)}\right)} \cdot \left(\boldsymbol{W}_V^{(l)}\boldsymbol{x}_j^{(l)}\right)$$

- where kernel $\kappa(\boldsymbol{x}, \boldsymbol{y}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) \rangle_{\mathcal{V}} \approx \phi(\boldsymbol{x})^T \phi(\boldsymbol{y})$
- Plug in

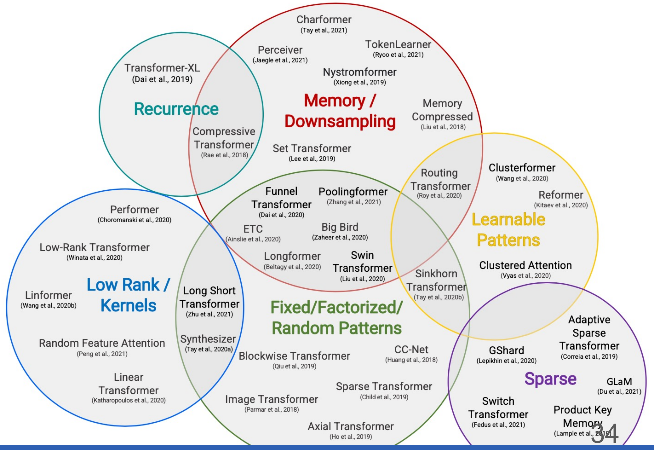
$$\boldsymbol{x}_i^{(l+1)} = \sum_{j=1}^{n} \frac{\phi(\boldsymbol{q}_i)^T \phi(\boldsymbol{k}_j)}{\sum_{k=1}^{n} \phi(\boldsymbol{q}_i)^T \phi(\boldsymbol{k}_k)} \cdot \boldsymbol{v}_j$$

$$= \frac{\left(\phi(\boldsymbol{q}_i)^T \boxed{\sum_{j=1}^{n} \phi(\boldsymbol{k}_j) \otimes \boldsymbol{v}_j}\right)^T}{\phi(\boldsymbol{q}_i)^T \boxed{\sum_{k=1}^{n} \phi(\boldsymbol{k}_k)}}.$$

VN in disguise

MPNN + VN w/ constant width & depth

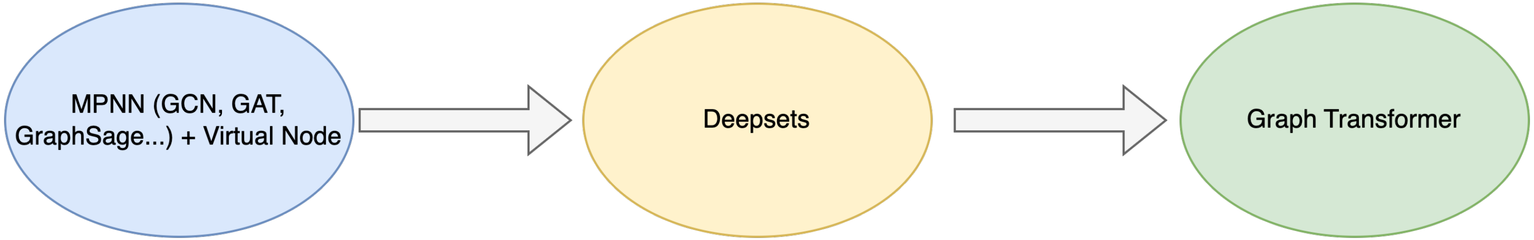
- Performer and Linear Transformer fall into such category
- Performer is used SOTA model GraphGPS
- They can be arbitrarily approximated by MPNN + VN
- There are many other ways to build linear transformer
    - Coarsening, shortcuts…
    - Unlikely MPNN + VN can approximate all of them



Choromanski et al. "Rethinking attention with performers." ICLR 2021
Katharopoulos et al. "Transformers are rnns: Fast autoregressive transformers with linear attention." ICML 2020.
Rampášek et al. "Recipe for a general, powerful, scalable graph transformer." NeurIPS 2022

Wide MPNN + VN

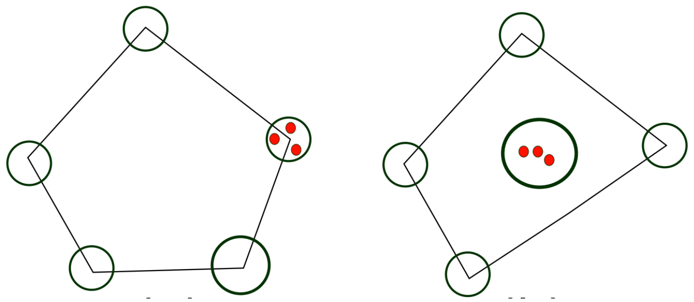
- Key observation: MPNN + VN can simulate equivariant DeepSets
- DeepSets layer: $L^{ds} = XA + \frac{1}{n}11^T XB + 1C^T$
- DeepSets is permutational equivariant universal
- Therefore, MPNN + VN is also permutational equivariant universal
- Therefore, MPNN + VN can approximate Transformer/SA layer
- Drawback: upper bound on width is $O(n^d)$



Segol, Nimrod, and Yaron Lipman. "On universal equivariant set networks." ICLR 2020

Deep MPNN + VN

- Need strong assumption on node features
- VN approximately selects (using attention) one node feature per iteration
- Do some computation and send message back to all nodes
- Repeat n rounds
- Assumption can be relaxed by allowing a more powerful attention mechanism (i.e. GATv2) in VN



Brody, Shaked, Uri Alon, and Eran Yahav. "How attentive are graph attention networks?." ICLR 2022

Exp1: MPNN + VN outperforms GT

- On Long Range Graph Benchmark (LRGB), it is observed that GT significantly outperforms MPNN
- We add VN and observe that MPNN + VN performs even better than GT

Model	# Params.	Peptides-func		Peptides-struct	
		Test AP before VN	Test AP after VN ↑	Test MAE before VN	Test MAE after VN ↓
GCN	508k	0.5930±0.0023	0.6623±0.0038	0.3496±0.0013	**0.2488±0.0021**
GINE	476k	0.5498±0.0079	0.6346±0.0071	0.3547±0.0045	0.2584±0.0011
GatedGCN	509k	0.5864±0.0077	0.6635±0.0024	0.3420±0.0013	0.2523±0.0016
GatedGCN+RWSE	506k	0.6069±0.0035	**0.6685±0.0062**	0.3357±0.0006	0.2529±0.0009
Transformer+LapPE	488k	0.6326±0.0126	-	0.2529±0.0016	-
SAN+LapPE	493k	0.6384±0.0121	-	0.2683±0.0043	-
SAN+RWSE	500k	0.6439±0.0075	-	0.2545±0.0012	-

Dwivedi, Vijay Prakash, et al. "Long range graph benchmark." NeurIPS 2022

Exp2: Stronger MPNN + VN implementation

Table 3: Test performance in graph-level OGB benchmarks (Hu et al., 2020). Shown is the mean ± s.d. of 10 runs.

Model	ogbg-molhiv	ogbg-molpcba	ogbg-ppa	ogbg-code2
	AUROC ↑	Avg. Precision ↑	Accuracy ↑	F1 score ↑
GCN	0.7606 ± 0.0097	0.2020 ± 0.0024	0.6839 ± 0.0084	0.1507 ± 0.0018
GCN+virtual node	0.7599 ± 0.0119	0.2424 ± 0.0034	0.6857 ± 0.0061	0.1595 ± 0.0018
GIN	0.7558 ± 0.0140	0.2266 ± 0.0028	0.6892 ± 0.0100	0.1495 ± 0.0023
GIN+virtual node	0.7707 ± 0.0149	0.2703 ± 0.0023	0.7037 ± 0.0107	0.1581 ± 0.0026
SAN	0.7785 ± 0.2470	0.2765 ± 0.0042	–	–
GraphTrans (GCN-Virtual)	–	0.2761 ± 0.0029	–	0.1830 ± 0.0024
K-Subtree SAT	–	–	0.7522 ± 0.0056	0.1937 ± 0.0028
GPS	0.7880 ± 0.0101	0.2907 ± 0.0028	0.8015 ± 0.0033	0.1894 ± 0.0024
MPNN + VN + NoPE	0.7676 ± 0.0172	0.2823 ± 0.0026	0.8055 ± 0.0038	0.1727 ± 0.0017
MPNN + VN + PE	0.7687 ± 0.0136	0.2848 ± 0.0026	0.8027 ± 0.0026	0.1719 ± 0.0013

Exp3: Forecasting sea surface temperature

- Discretize regions of interest as graphs
- Run MPNN + VN / GT for time series forecasting
- Observe MPNN + VN improves MPNN, and outperforms Linear Transformer
- Still fall behind TF-Net, a SOTA method for spatiotemporal forecasting

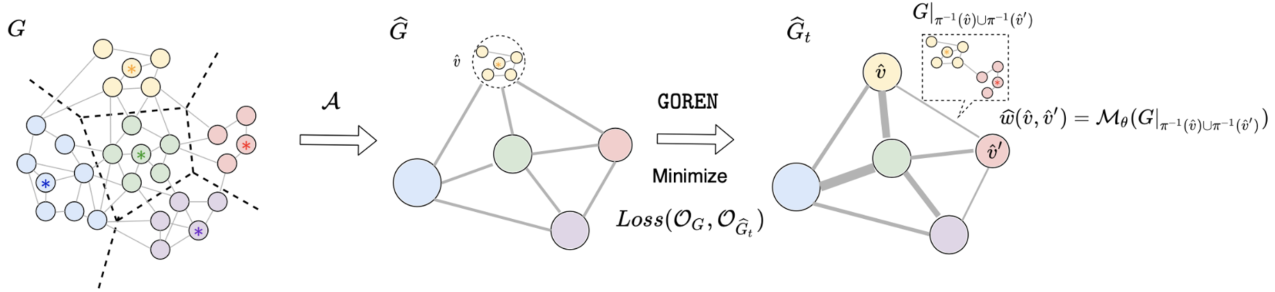
Table 5: Results of SST prediction.

Model	**4 weeks**	**2 weeks**	**1 week**
MLP	0.3302	0.2710	0.2121
TF-Net	0.2833	**0.2036**	**0.1462**
Linear Transformer + LapPE	0.2818	0.2191	0.1610
MPNN	0.2917	0.2281	0.1613
MPNN + VN	**0.2806**	0.2130	0.1540

Wang, Rui, et al. "Towards physics-informed deep learning for turbulent flow prediction." KDD 2020.

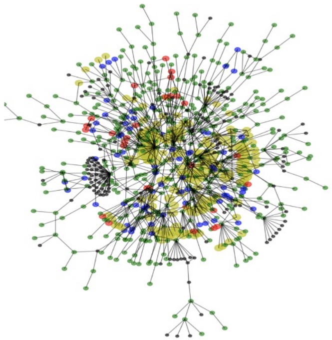
Graph Coarsening with Neural Networks

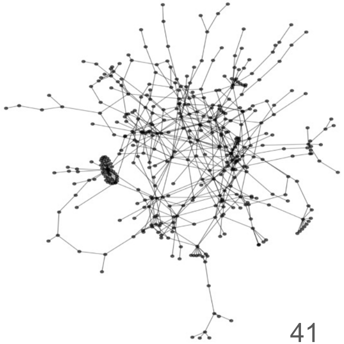
Chen Cai, Dingkang Wang, Yusu Wang
ICLR 2021



Motivation

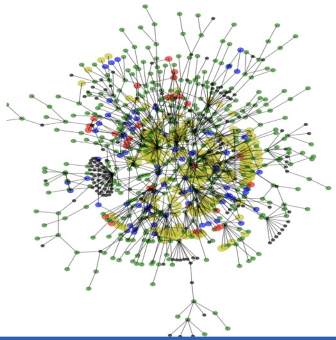
- Make a small graph out of a large graph while preserving some properties
- Fundamental operation
- Sister problem of edge sparsification by Spielman & Teng
- Useful for visualization, scientific computation, and other downstream tasks

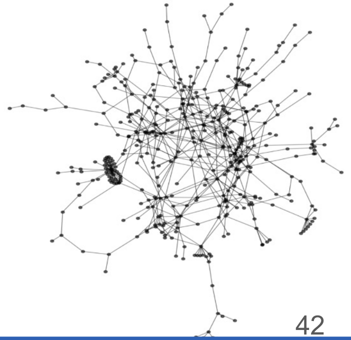




Key questions

- What properties are we trying to preserve?
  - Spectral property
  - Need to define operators, projection and lift map
- Edge weight optimization
  - Most algorithms do not optimize edge weights
  - Observation: optimizing edge weights brings significant improvements
- How to assign edge weights (GNN)
  - Subgraph regression
  - Good generalization





Graph coarsening

- We can not preserve everything in general. So what properties are we considering?
- Spectral property!

$$G \underset{\mathcal{U}}{\overset{\mathcal{P}}{\rightleftarrows}} \hat{G}$$

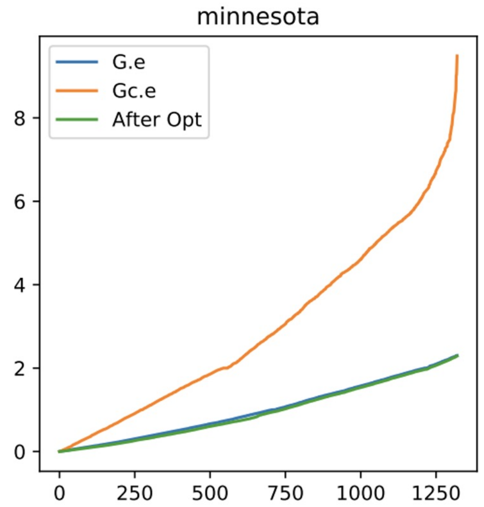
- Define projection/lift operator and their properties

Invariants under lift operator

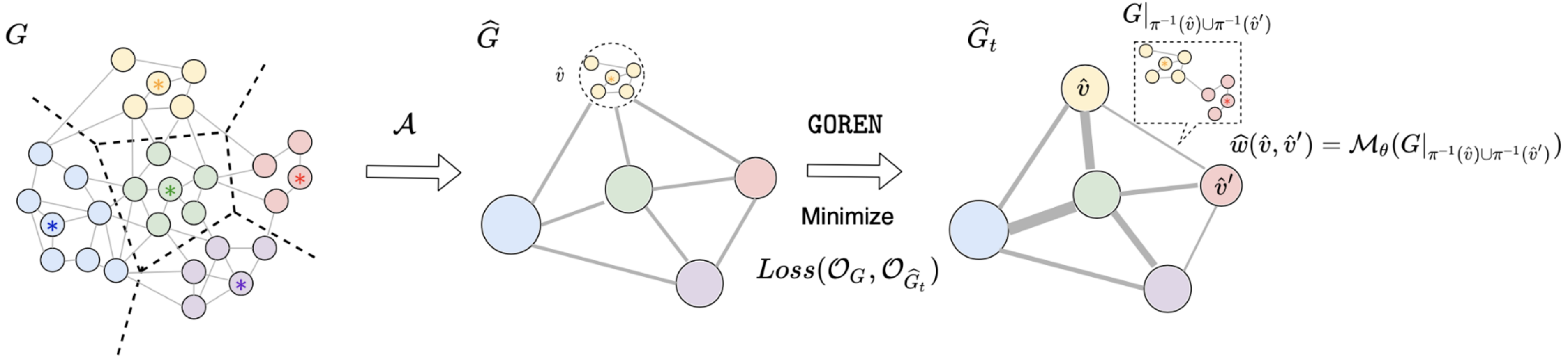
Quantity $\mathcal{F}$ of interest	$\mathcal{O}_G$	Projection $\mathcal{P}$	Lift $\mathcal{U}$	$\mathcal{O}_{\widehat{G}}$	Invariant under $\mathcal{U}$
Quadratic form Q	L	P	P^+	Combinatorial Laplace $\widehat{L}$	$\mathsf{Q}_L(\mathcal{U}\hat{x}) = \mathsf{Q}_{\widehat{L}}(\hat{x})$
Rayleigh quotient R	L	$\Gamma^{-1/2}(P^+)^T$	$P^+\Gamma^{-1/2}$	Doubly-weighted Laplace $\widehat{\mathsf{L}}$	$\mathsf{R}_L(\mathcal{U}\hat{x}) = \mathsf{R}_{\widehat{\mathsf{L}}}(\hat{x})$
Quadratic form Q	$\mathcal{L}$	$\widehat{D}^{1/2}PD^{-1/2}$	$D^{1/2}(P^+)\widehat{D}^{-1/2}$	Normalized Laplace $\widehat{\mathcal{L}}$	$\mathsf{Q}_{\mathcal{L}}(\mathcal{U}\hat{x}) = \mathsf{Q}_{\widehat{\mathcal{L}}}(\hat{x})$

Key observation

- Existing coarsening algorithm does not optimize for edge weight
- Theory: iterative algorithm with convergence property
- Practice: nearly identical eigenvalues alignment after optimization
- So let’s learn the edge weight
  - cvx. slow and does not generalize
  - neural network: suboptimal but generalize



Graph cOarsening RefinemEnt Network (GOREN)



Experiments

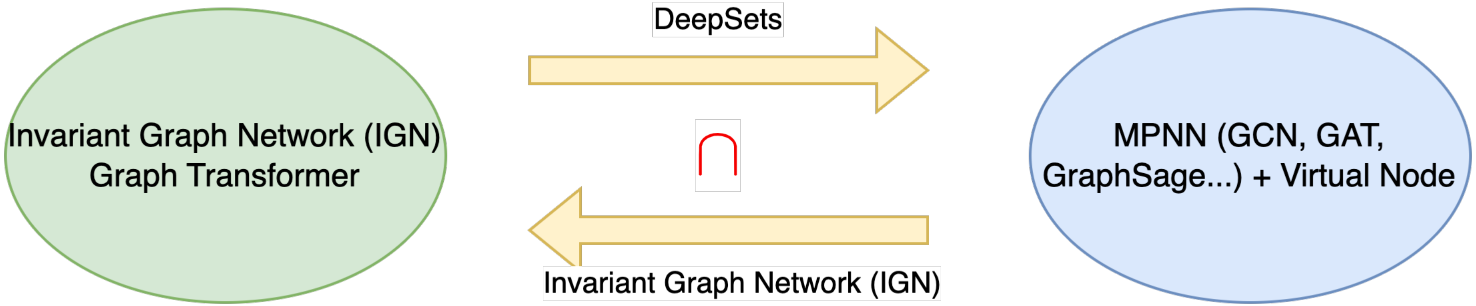
- Extensive experiments on synthetic graphs and real networks
- Synthetic graphs from common generative models
- Real networks: shape meshes; citation networks; largest one has 89k nodes

Table 3: Loss: quadratic loss. Laplacian: combinatorial Laplacian for both original and coarse graphs. Each entry $x(y)$ is: $x =$ loss w/o learning, and $y =$ improvement percentage.

	Dataset	BL	Affinity	Algebraic Distance	Heavy Edge	Local var (edges)	Local var (neigh.)
Synthetic	BA	0.44 (16.1%)	0.44 (4.4%)	0.68 (4.3%)	0.61 (3.6%)	0.21 (14.1%)	0.18 (72.7%)
	ER	0.36 (1.1%)	0.52 (0.8%)	0.35 (0.4%)	0.36 (0.2%)	0.18 (1.2%)	0.02 (7.4%)
	GEO	0.71 (87.3%)	0.20 (57.8%)	0.24 (31.4%)	0.55 (80.4%)	0.10 (59.6%)	0.27 (65.0%)
	WS	0.45 (62.9%)	0.09 (82.1%)	0.09 (60.6%)	0.52 (51.8%)	0.09 (69.9%)	0.11 (84.2%)
Real	CS	0.39 (40.0%)	0.21 (29.8%)	0.17 (26.4%)	0.14 (20.9%)	0.06 (36.9%)	0.0 (59.0%)
	Flickr	0.25 (10.2%)	0.25 (5.0%)	0.19 (6.4%)	0.26 (5.6%)	0.11 (11.2%)	0.07 (21.8%)
	Physics	0.40 (47.4%)	0.37 (42.4%)	0.32 (49.7%)	0.14 (28.0%)	0.15 (60.3%)	0.0 (-0.3%)
	PubMed	0.30 (23.4%)	0.13 (10.5%)	0.12 (15.9%)	0.24 (10.8%)	0.06 (11.8%)	0.01 (36.4%)
	Shape	0.23 (91.4%)	0.08 (89.8%)	0.06 (82.2%)	0.17 (88.2%)	0.04 (80.2%)	0.08 (79.4%)

Conclusion

- Local-to-Global Perspectives on GNN
- Two works on theory of global GNN
  - Convergence of IGN (global GNN)
  - Connection between MPNN and GT (connection)
- One applied work:
  - Graph Coarsening with Neural Networks local GNN)



Future direction

- **Expressivity** research needs to go beyond connectivity and model 3d positions & node features
- Harder question: **optimization and generalization** of GNN
- Equivariant GNN + Diffusion for **conditional generation** of structured data
- Geometric/topological tools to understand the **regularity** of molecule/material spaces & **hardness** of learning/sampling

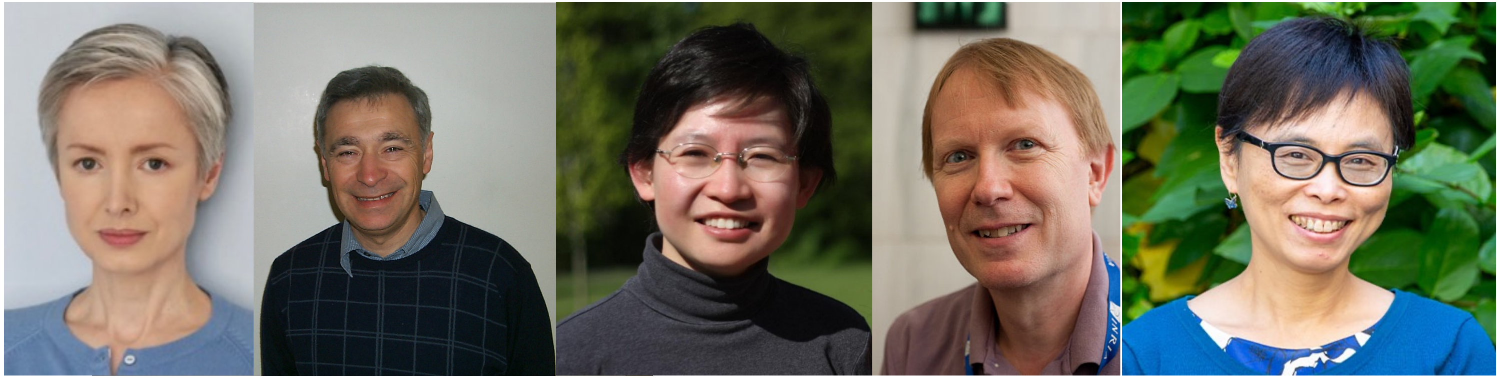
Acknowledgement

Bahador Bahmani (Columbia)
Goonmeet Bajaj (OSU)
Gopinath Balamurugan (Universität Tübingen)
Beatrice Bevilacqua (Predue)
Michael M. Bronstein (Oxford & DeepMind)
Rafael Gómez-Bombarelli (MIT)
Saket Gurukar (OSU)
Fabrizio Frasca (ICL)
Truong Son Hy (UCSD & IST)
Woojin Kim (Duke)
Moniba Keymanesh (IIT)
Saravana Kumar (IIT)
Derek Lim (MIT)

Facundo Memoli (OSU)
Benjamin Kurt Miller (U of Amsterdam)
Anasua Mitra (IIT)
Haggai Maron (Nvidia & Technion)
Pranav Maneriker (OSU)
Lucas Magee (UCSD)
Ran Ma (Columbia)
Vedang Patel (OSU)
Srinivasan Parthasarathy (OSU)
Aakash Srinivasan (IIT)
Tess Smidt (MIT)
Balaraman Ravindran (IIT)
Jian Tang (MILA)

Priyesh Vijayan (IIT)
Nikolaos Vlassis (Columbia)
Wujie Wang (MIT)
Yusu Wang (UCSD)
Teng-Fong Wong (Stony Brook)
Dingkang Wang (OSU)
Zeyu Xiong (Columbia)
Rose Yu (UCSD)
Jie Zhang (IIT)

Acknowledgement



Thank You!
Questions?