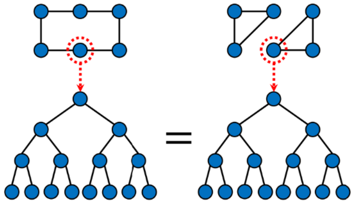
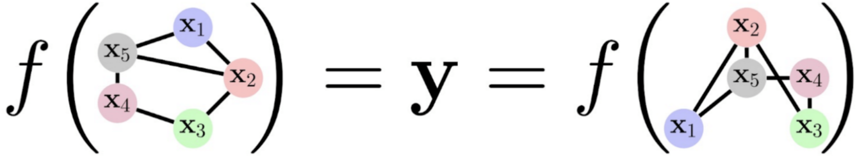
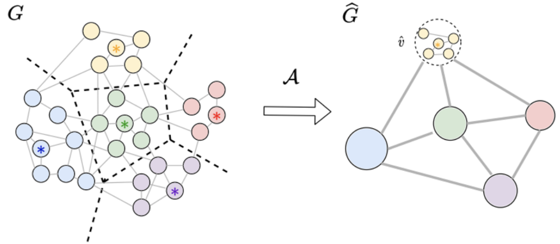
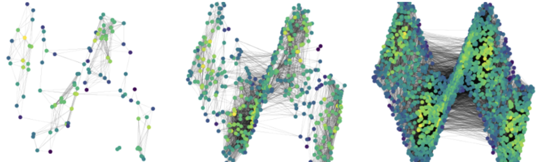
Learning Over Space of Graphs with Neural Networks

Chen Cai

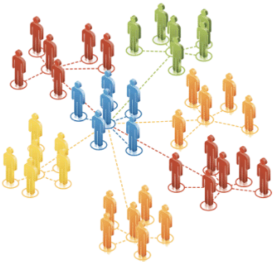






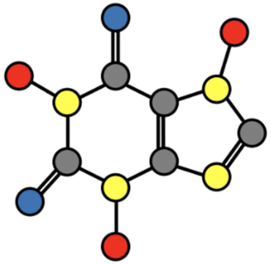


Graphs

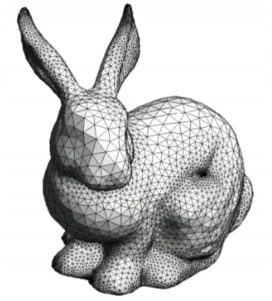


Social networks

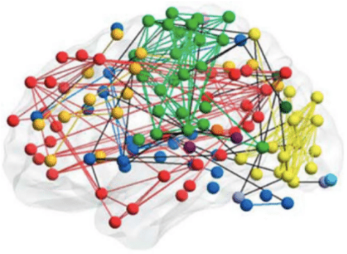
Interaction networks



Molecules



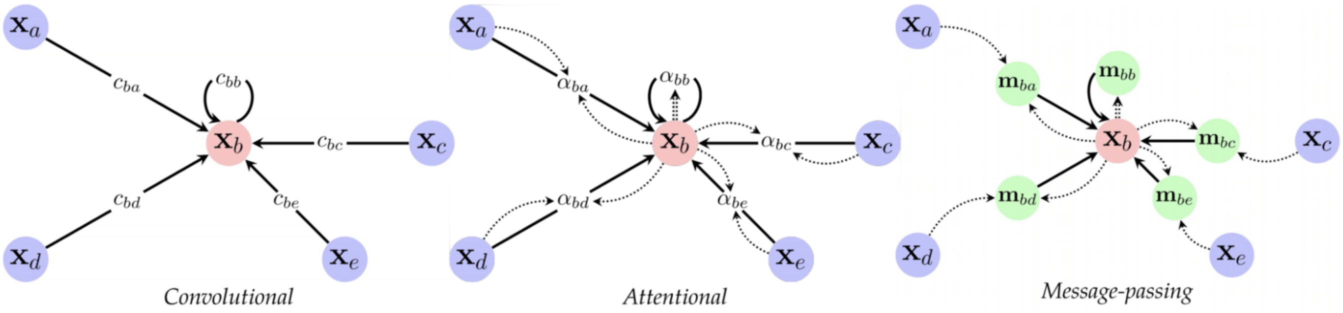
Meshes



Functional networks

Graph Neural Network

- Generalize CNN to graphs
- Permutation equivariant/invariant $f(P^T AP) = P^T f(A)P / f(P^T AP) = f(A)$
- Handles rich node/edge scalar/vector/high-order tensor features
- Train on small graphs, generalize to large graphs



$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} c_{ij}\psi(\mathbf{x}_j)\right) \qquad \mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} a(\mathbf{x}_i, \mathbf{x}_j)\psi(\mathbf{x}_j)\right) \qquad \mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} \psi(\mathbf{x}_i, \mathbf{x}_j)\right)$$

My Research Topics in GNN

Theory

- Expressive power of GNN & Universality [1]
- **Convergence and stability [2] (a sequence of graphs)**
- Over-Smoothing for GNN [3]
- Hardness of learning combinatorial optimization problems with GNN (ongoing)

Applications

- **Graph coarsening [4] (large graph → small graph)**
- **Molecule modeling [5] (small graph → large graph)**

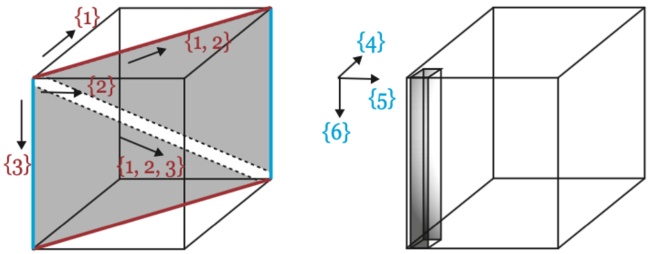
[1] Equivariant Subgraph Aggregation Networks
[2] Convergence of Invariant Graph Networks
[3] A Note on Over-Smoothing for Graph Neural Networks
[4] Graph Coarsening with Neural Networks
[5] Generative Coarse-Graining of Molecular Conformations

Convergence of Invariant Graph Networks

Chen Cai & Yusu Wang

Arxiv 2022, under submission

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



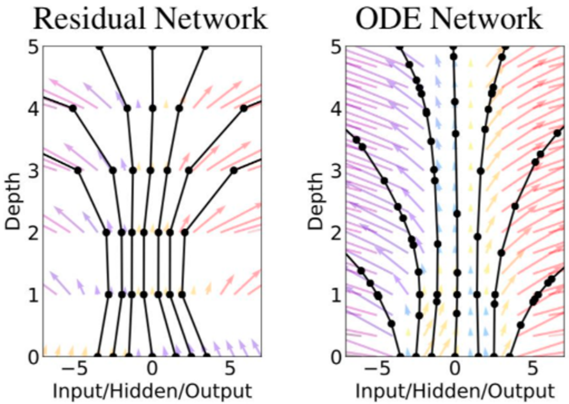
Convergence in Deep Learning

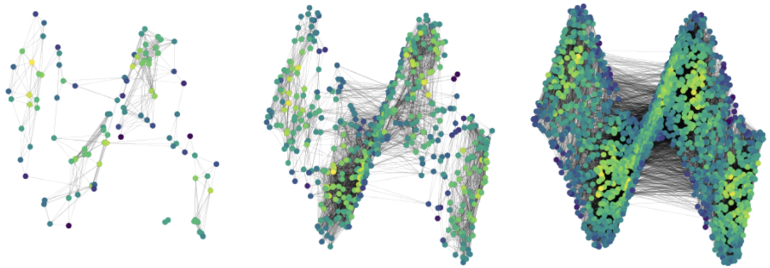
- Increase width: Neural Tangent Kernel
- Increase depth: Neural ODE
- Increase input size? Convergence of graph neural network!

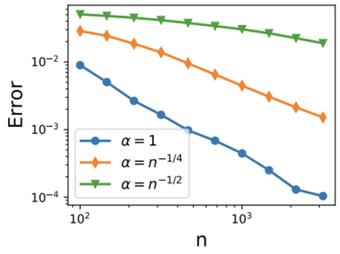
depth

$$\Theta^{(L)}(x,y) = \sum_{p=1}^{P} \frac{d}{d\theta_p} f_\theta(x) \, \frac{d}{d\theta_p} f_\theta(y)$$

two samples   all parameters

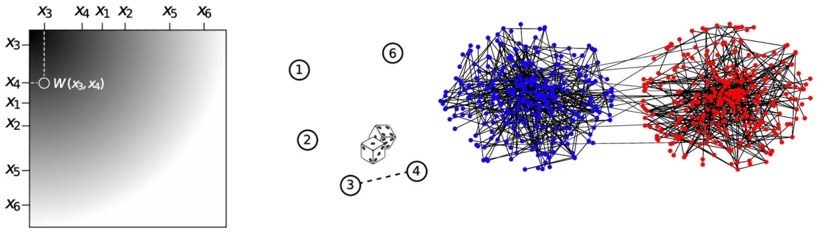




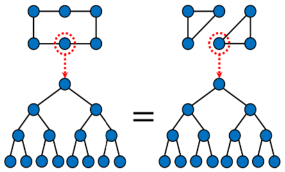


Setup & Existing work

- Model
  - graphon $W: [0,1]^2 \to [0,1]$
  - edge probability discrete model
  - edge weight continuous model
- Mainly study 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) under 1) edge probability discrete model and 2) edge weight continuous model

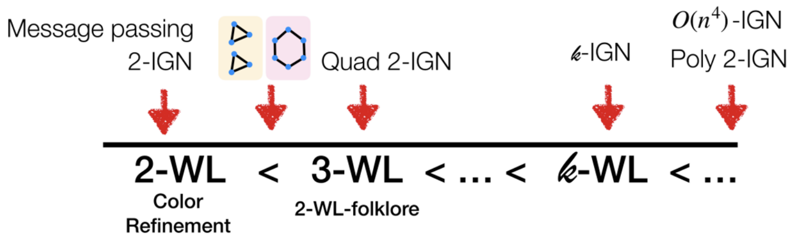
Invariant Graph Network (IGN)

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

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

Invariant Graph Network (IGN)

- $F = h \circ L^{(T)} \circ \sigma \cdots \circ \sigma \circ L^{(1)}$
- Depending on order of intermediate tensor, 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
  - As k increase, k-IGN reaches universality



Operations	Discrete	Continuous	Partitions
1-2: The identity and transpose operations	$T(A) = A$ $T(A) = A^T$	$T(W) = W$ $T(W) = W^T$	$\{\{1,3\},\{2,4\}\}$ $\{\{1,4\},\{2,3\}\}$
3: The diag operation	$T(A) = \mathrm{Diag}(\mathrm{Diag}^*(A))$	$T(W)(u,v) = W(u,v)\mathrm{I}_{u=v}$	$\{\{1,2,3,4\}\}$
4-6: Average of rows replicated on rows/ columns/ diagonal	$T(A) = \frac{1}{n}A\mathbf{1}\mathbf{1}^T$ $T(A) = \frac{1}{n}\mathbf{1}(A\mathbf{1})^T$ $T(A) = \frac{1}{n}\mathrm{Diag}(A\mathbf{1})$	$T(W)(u,*) = \int W(u,v)dv$ $T(W)(*,u) = \int W(u,v)dv$ $T(W)(u,v) = \mathrm{I}_{u=v}\int W(u,v')dv'$	$\{\{1,4\},\{2\},\{3\}\}$ $\{\{1,3\},\{2\},\{4\}\}$ $\{\{1,3,4\},\{2\}\}$
7-9: Average of columns replicated on rows/ columns/ diagonal	$T(A) = \frac{1}{n}A^T\mathbf{1}\mathbf{1}^T$ $T(A) = \frac{1}{n}\mathbf{1}(A^T\mathbf{1})^T$ $T(A) = \frac{1}{n}\mathrm{Diag}(A^T\mathbf{1})$.	$T(W)(*,v) = \int W(u,v)du$ $T(W)(v,*) = \int W(u,v)du$ $T(W)(u,v) = \mathrm{I}_{u=v}\int W(u',v)du'$	$\{\{1\},\{2,4\},\{3\}\}$ $\{\{1\},\{2,3\},\{4\}\}$ $\{\{1\},\{2,3,4\}\}$
10-11: Average of all elements replicated on all matrix/ diagonal	$T(A) = \frac{1}{n^2}(\mathbf{1}^TA\mathbf{1})\cdot\mathbf{1}\mathbf{1}^T$ $T(A) = \frac{1}{n^2}(\mathbf{1}^TA\mathbf{1})\cdot\mathrm{Diag}(\mathbf{1})$.	$T(W)(*,*) = \int W(u,v)dudv$ $T(W)(u,v) = \mathrm{I}_{u=v}\int W(u',v')du'dv'$	$\{\{1\},\{2\},\{3\},\{4\}\}$ $\{\{1\},\{2\},\{3,4\}\}$
12-13: Average of diagonal elements replicated on all matrix/diagonal	$T(A) = \frac{1}{n}(\mathbf{1}^T\mathrm{Diag}^*(A))\cdot\mathbf{1}\mathbf{1}^T$ $T(A) = \frac{1}{n}(\mathbf{1}^T\mathrm{Diag}^*(A))\cdot\mathrm{Diag}(\mathbf{1})$	$T(W)(*,*) = \int \mathrm{I}_{u=v}W(u,v)dudv$ $T(W)(u,v) = \mathrm{I}_{u=v}\int W(u',u')du'$	$\{\{1,2\},\{3\},\{4\}\}$ $\{\{1,2\},\{3,4\}\}$
14-15: Replicate diagonal elements on rows/columns	$T(A) = \mathrm{Diag}^*(A)\mathbf{1}^T$ $T(A) = \mathbf{1}\mathrm{Diag}^*(A)^T$	$T(W)(u,v) = W(u,u)$ $T(W)(u,v) = W(v,v)$	$\{\{1,2,4\},\{3\}\}$ $\{\{1,2,3\},\{4\}\}$

2-IGN

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

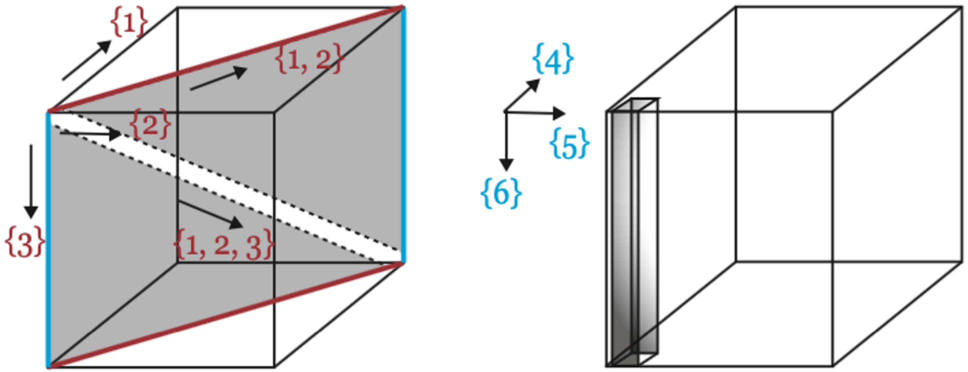
Definition (partition norm): The partition norm of 2-tensor $A \in \mathbb{R}^{n^2}$ is defined as $\|A\|_{pn} := \left(\frac{Diag^*(A)}{\sqrt{n}}, \frac{\|A\|_2}{n}\right)$. The continuous analog of the partition-norm for graphon $W \in \mathcal{W}$ is defined as $\|W\|_{pn} := (\sqrt{\int W^2(u,u)du},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

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

Space of linear (permutation) equivariant maps

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

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



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

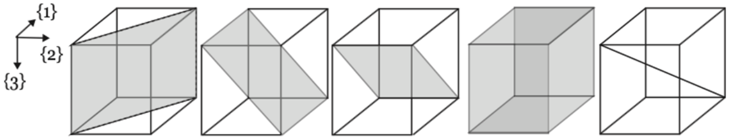
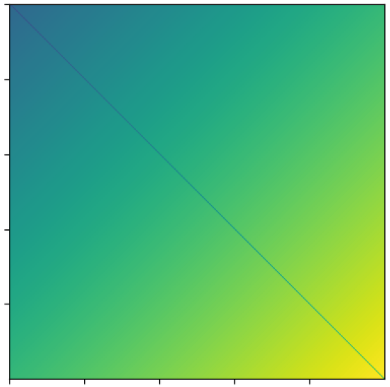
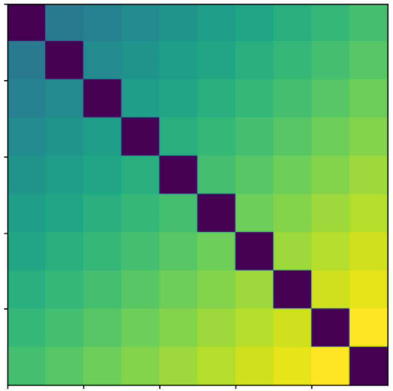


Figure 1: Five possible "slices" of a 3-tensor, corresponding to bell(3) = 5 paritions of [3]. From left to right: a) $\{\{1,2\},\{3\}\}$ b) $\{\{1\},\{2,3\}\}$ c) $\{\{1,3\},\{2\}\}$ d) $\{\{1\},\{2\},\{3\}\}$ e) $\{\{1,2,3\}\}$.

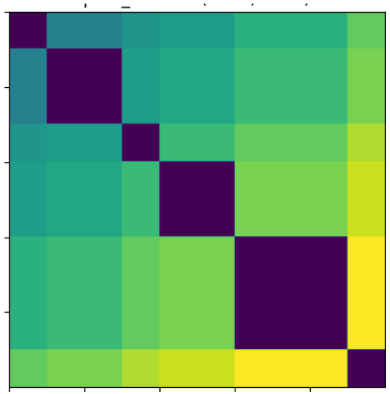
Edge Weight Continuous Model



W



W_n



$\widetilde{W}_n$

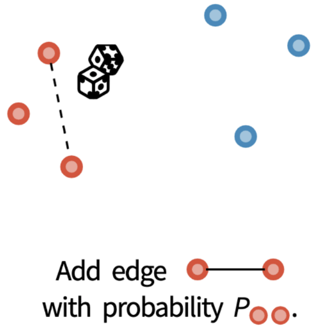
$cIGN(W_n) \to cIGN(W)$

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

Edge Probability Discrete Model

$$RMSE_U(\phi_c(W), \phi_d(A_{n\times n}))$$

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



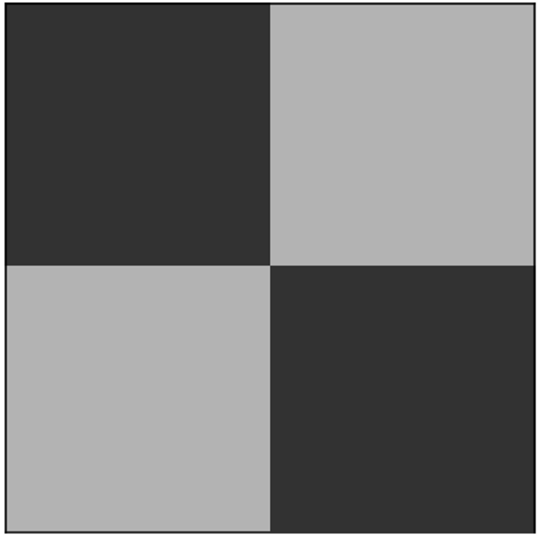
$$RMSE_U(f, x) := \left\| S_u f - \frac{x}{n} \right\|_2 = \left(n^{-2} \sum_{i=0}^{n} \sum_{j=0}^{n} \left\| f(u_i, u_j) - x(i,j) \right\|^2 \right)^{1/2}$$

Negative Result

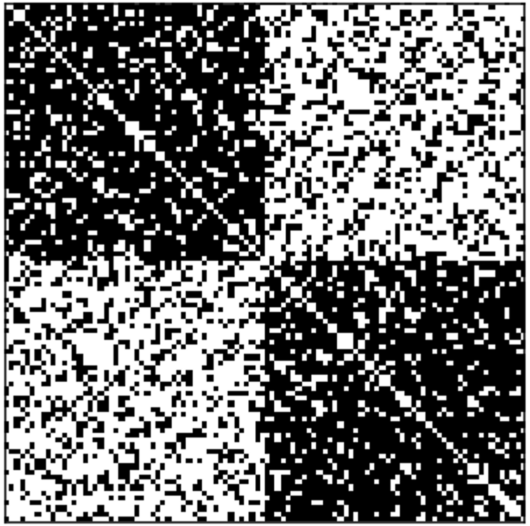
Informal Theorem (negative result) [Cai & Wang, 2022]

Under mild assumptions on W, given any IGN architecture, there exists a set of parameter θ such that the convergence of IGN_θ to $cIGN_\theta$ 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.

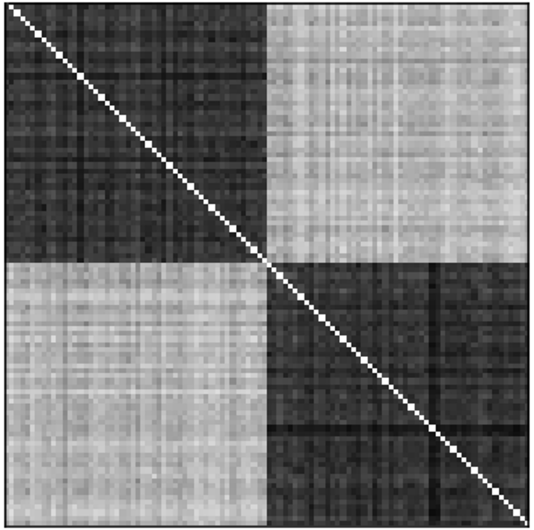
Edge Probability Estimation



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 $\frac{1}{n}\left\|W - \widehat{W}\right\|_2$ converges to 0 in probability. Let Φ_c, Φ_d be continuous and discrete IGN-small. Then $RMSE_U(\phi_c([W, Diag(X)]), \phi_d\left(\left[\widehat{W_{n\times n}}, Diag(\widetilde{X_n})\right]\right)$ converges to 0 in probability.

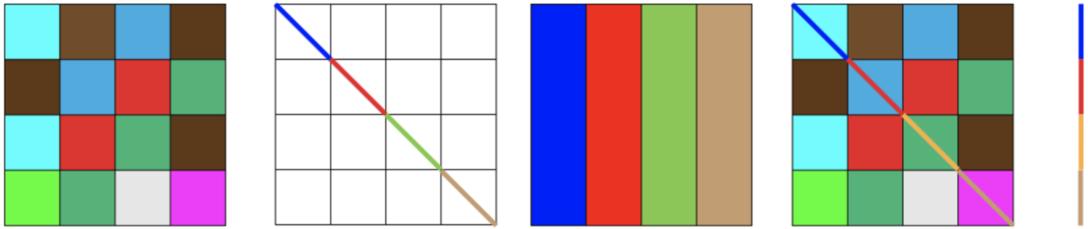
- Proof leverages
  - Statistical guarantee of edge smoothing
  - Property of basis elements of k-IGN
  - Standard algebraic manipulation
  - Property of sampling operator

$$
\begin{aligned}
&\mathrm{RMSE}_U(\Phi_c(W), \Phi_d(\widehat{W}_{n\times n})) \\
&= \|S_U\Phi_c(W) - \frac{1}{\sqrt{n}}\Phi_d(\widehat{W}_{n\times n})\| \\
&\leqslant \underbrace{\|S_U\Phi_c(W) - S_U\Phi_c(\widetilde{W_n})\|}_{\text{First term: discrization error}} + \underbrace{\|S_U\Phi_c(\widetilde{W_n}) - \Phi_d S_U(\widetilde{W_n})\|}_{\text{Second term: sampling error}} \\
&+ \underbrace{\|\Phi_d S_U(\widetilde{W_n}) - \frac{1}{\sqrt{n}}\Phi_d(\widehat{W}_{n\times n})\|}_{\text{Third term: estimation error}} \qquad (8)
\end{aligned}
$$

IGN-small

- A subset of IGN

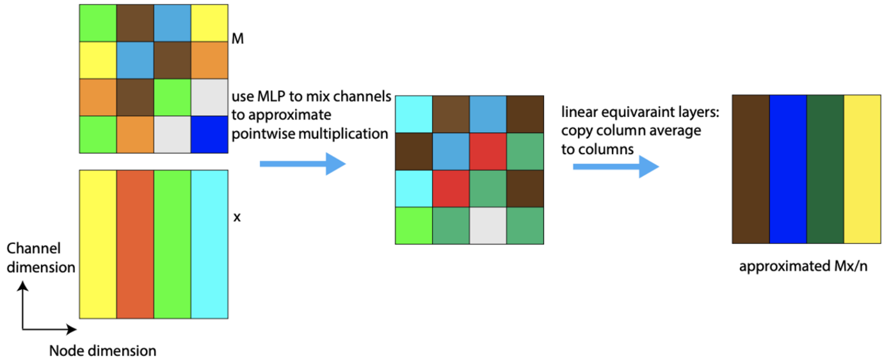
Definition (IGN-small): Let $\widetilde{W_{n,E}}$ be a graphon with "chessboard pattern", i.e., it is a piecewise constant graphon where each block is of the same size. Similarly, define $\widetilde{X_{n,E}}$ as the 1D analog. IGN-small denotes a subset of IGN that satisfies $S_n \phi_c([\widetilde{W_{n,E}},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



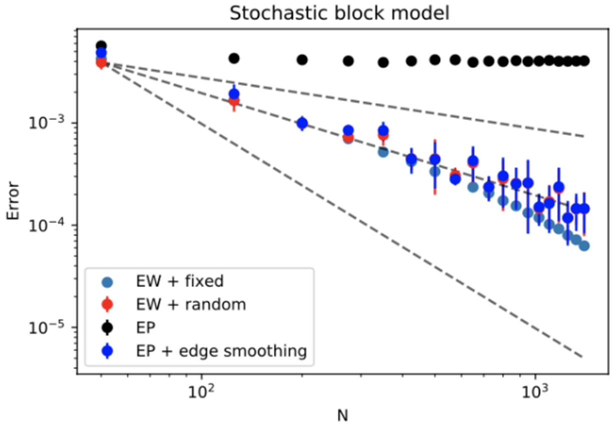
(a) (b) (c) (d) (e)

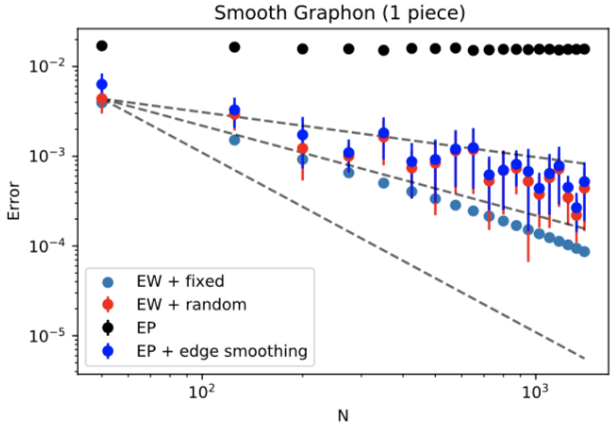
IGN-small Can Approximate SGNN Arbitrarily Well

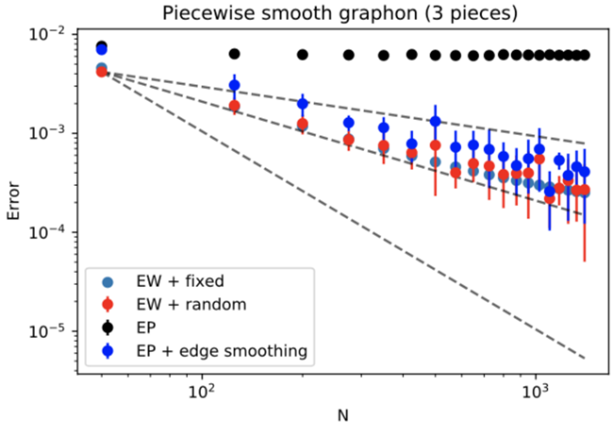
- Spectral GNN (SGNN) $z_j^{(l+1)} = \rho(\sum_{i=1}^{d_l} h_{ij}^{(l)}(L) z_i^{(l)} + b_j^{(l)} 1_n)$
- Main GNN considered in convergence literature
- Proof idea:
  - It suffice to approximate Lx
  - 2-IGN basis functions can compute L and do matrix-vector multiplication



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

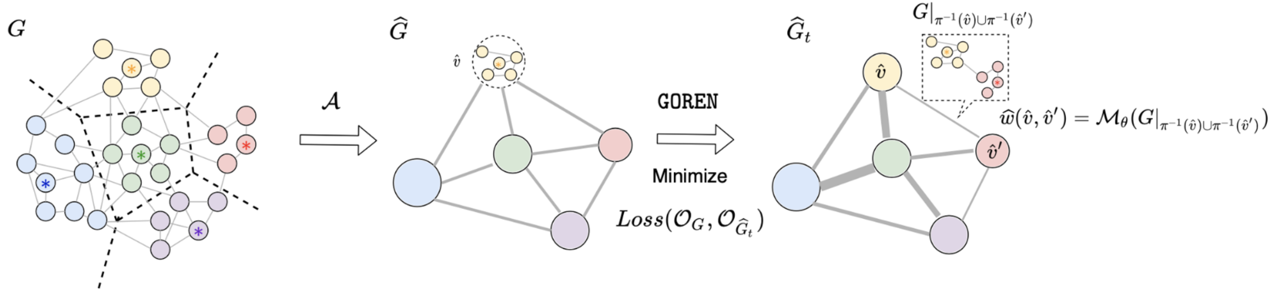
Edge probability discrete model

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

Graph Coarsening with Neural Networks

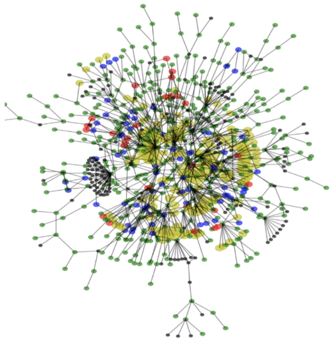
Chen Cai, Dingkang Wang, Yusu Wang

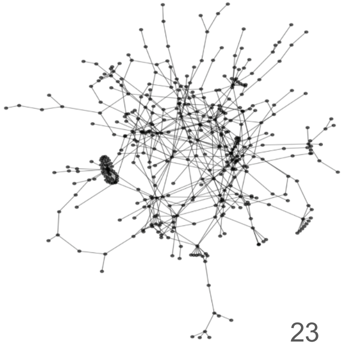
ICLR 2021



Graph Coarsening: Motivation

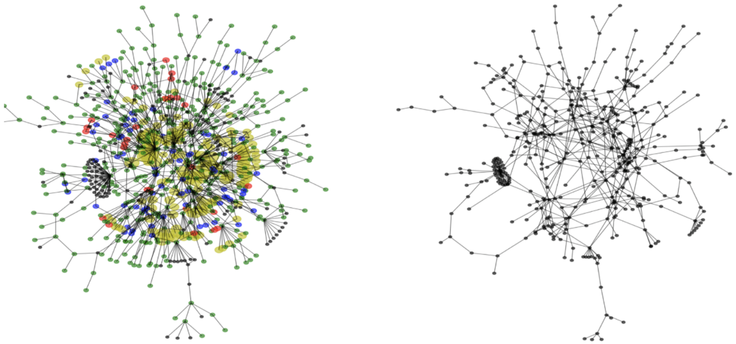
- Make a small graph out of a large graph while preserving some properties
- Fundamental operation
- Useful for visualization, scientific computing, and other downstream tasks
- Edge sparsification algorithm (Spielman & Teng)





Agenda & Key Questions

- What properties are we trying to preserve?
  - Spectral property
  - Need to define operators on original and coarse graph (double weighted Laplacian)
- 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

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

$$G \underset{\mathcal{P}}{\overset{\mathcal{U}}{\leftrightarrows}} \widehat{G}$$

- Define projection/lift operator; operator of interest; and their properties

Laplace Operator

- Laplacian on graphs: $L = D - W$
- Normalized Laplacian: $\mathcal{L} = D^{-1/2} L D^{-1/2}$
- Discrete analog of Laplace operator
- Used in spectral thoery, diffusion process, image processing…

How to Measure the Quality?

- Compare $\mathcal{F}(\mathcal{O}_G, f)$ and $\mathcal{F}\left(\mathcal{O}_{\hat{G}}, \hat{f}\right)$

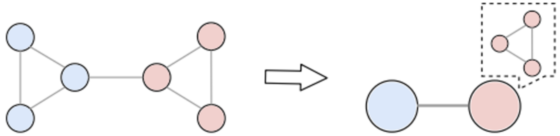
- $\mathcal{F}$ can be quadratic form $x^T L x$ or Rayleigh quotient $\frac{x^T L\, x}{x^T x}$
- $\mathcal{O}_G, \mathcal{O}_{\hat{G}}$ are the Laplace operators
- f is graph signal such as the eigenvector of graph Laplacian

Example

$$\mathcal{P} = P = \begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 1/3 \end{bmatrix} \quad \mathcal{U} = P^+ = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Proposition A.2. *For any vector $\hat{x} \in \mathbb{R}^n$, we have that $\mathsf{Q}_{\widehat{L}}(\hat{x}) = \mathsf{Q}_L(P^+\hat{x})$. In other words, set $x := P^+\hat{x}$ as the lift of $\hat{x}$ in $\mathbb{R}^N$, then $\hat{x}^T\widehat{L}\hat{x} = x^T L x$.*

Proof. $\mathsf{Q}_L(\mathcal{U}\hat{x}) = (\mathcal{U}\hat{x})^T L\mathcal{U}\hat{x} = \hat{x}(P^+)^T L P^+\hat{x}^T = \hat{x}^T\widehat{L}\hat{x} = \mathsf{Q}_{\widehat{L}}(\hat{x})$ □

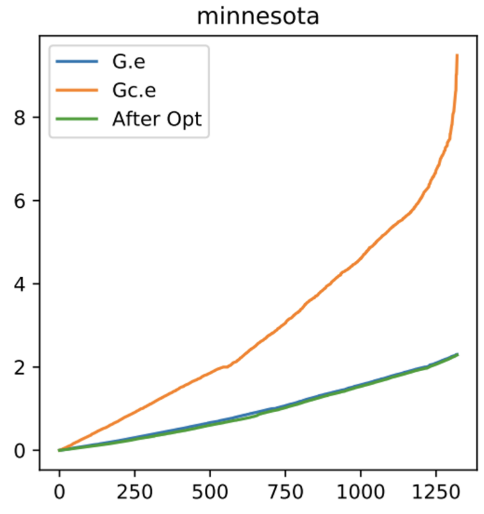


Invariants under Lift

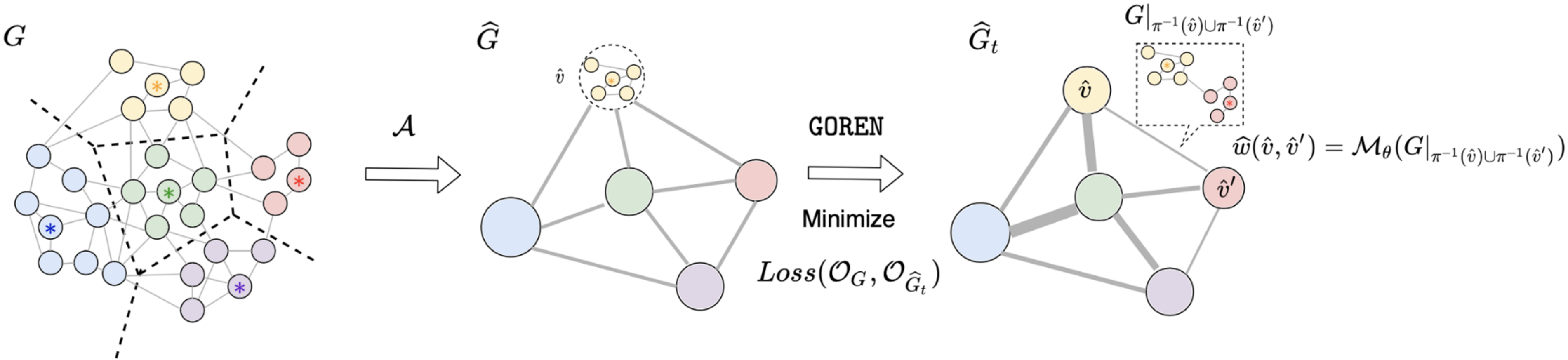
Quantity $\mathcal{F}$ of interest	$\mathcal{O}_G$	Projection $\mathcal{P}$	Lift $\mathcal{U}$	$\mathcal{O}_{\widehat{G}}$	Invariant under $\mathcal{U}$
Quadratic form Q	L	P	P^+	Combinatorial Laplace $\widehat{L}$	$\mathsf{Q}_L(\mathcal{U}\hat{x}) = \mathsf{Q}_{\widehat{L}}(\hat{x})$
Rayleigh quotient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



Model Details

- Simple feature based on node degree
- Graph Isomorphism Network (GIN)
- Generic optimization (Adam with constant learning rate)
- No bells and whistles

Experiment: Proof of Concept

Table 2: The error reduction after applying GOREN.

Dataset	Affinity	Algebraic Distance	Heavy Edge	Local var (edges)	Local var (neigh.)
Airfoil	91.7%	88.2%	86.1%	43.2%	73.6%
Minnesota	49.8%	57.2%	30.1%	5.50%	1.60%
Yeast	49.7%	51.3%	37.4%	27.9%	21.1%
Bunny	84.7%	69.1%	61.2%	19.3%	81.6%

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%)

Experiment: Generalization

- Generalize to graph from same generative model
- Train on small subgraph, generalize to much large (25x) graphs
- Works for different objective functions

Table 4: Loss: quadratic loss. Laplacian: normalized Laplacian for 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	0.13 (76.2%)	0.14 (45.0%)	0.15 (51.8%)	0.15 (46.6%)	0.14 (55.3%)	0.06 (57.2%)
	ER	0.10 (82.2%)	0.10 (83.9%)	0.09 (79.3%)	0.09 (78.8%)	0.06 (64.6%)	0.06 (75.4%)
	GEO	0.04 (52.8%)	0.01 (12.4%)	0.01 (27.0%)	0.03 (56.3%)	0.01 (-145.1%)	0.02 (-9.7%)
	WS	0.05 (83.3%)	0.01 (-1.7%)	0.01 (38.6%)	0.05 (50.3%)	0.01 (40.9%)	0.01 (10.8%)
Real	CS	0.08 (58.0%)	0.06 (37.2%)	0.04 (12.8%)	0.05 (41.5%)	0.02 (16.8%)	0.01 (50.4%)
	Flickr	0.08 (-31.9%)	0.06 (-27.6%)	0.06 (-67.2%)	0.07 (-73.8%)	0.02 (-440.1%)	0.02 (-43.9%)
	Physics	0.07 (47.9%)	0.06 (40.1%)	0.04 (17.4%)	0.04 (61.4%)	0.02 (-23.3%)	0.01 (35.6%)
	PubMed	0.05 (47.8%)	0.05 (35.0%)	0.05 (41.1%)	0.12 (46.8%)	0.03 (-66.4%)	0.01 (-118.0%)
	Shape	0.02 (84.4%)	0.01 (67.7%)	0.01 (58.4%)	0.02 (87.4%)	0.0 (13.3%)	0.01 (43.8%)

Experiments: non-differentiable objective

- The eigenvalue alignment is non-differentiable w.r.t weights
- Use Rayleigh quotient as a proxy
- More challenging

Table 5: Loss: Eigenerror. Laplacian: combinatorial Laplacian for original graphs and doubly-weighted Laplacian for coarse ones. Each entry $x(y)$ is: $x =$ loss w/o learning, and $y =$ improvement percentage. † stands for out of memory.

	Dataset	BL	Affinity	Algebraic Distance	Heavy Edge	Local var (edges)	Local var (neigh.)
Synthetic	BA	0.36 (7.1%)	0.17 (8.2%)	0.22 (6.5%)	0.22 (4.7%)	0.11 (21.1%)	0.17 (-15.9%)
	ER	0.61 (0.5%)	0.70 (1.0%)	0.35 (0.6%)	0.36 (0.2%)	0.19 (1.2%)	0.02 (0.8%)
	GEO	1.72 (50.3%)	0.16 (89.4%)	0.18 (91.2%)	0.45 (84.9%)	0.08 (55.6%)	0.20 (86.8%)
	WS	1.59 (43.9%)	0.11 (88.2%)	0.11 (83.9%)	0.58 (23.5%)	0.10 (88.2%)	0.12 (79.7%)
Real	CS	1.10 (18.0%)	0.55 (49.8%)	0.33 (60.6%)	0.42 (44.5%)	0.21 (75.2%)	0.0 (-154.2%)
	Flickr	0.57 (55.7%)	†	0.33 (20.2%)	0.31 (55.0%)	0.11 (67.6%)	0.07 (60.3%)
	Physics	1.06 (21.7%)	0.58 (67.1%)	0.33 (69.5%)	0.35 (64.6%)	0.20 (79.0%)	0.0 (-377.9%)
	PubMed	1.25 (7.1%)	0.50 (15.5%)	0.51 (12.3%)	1.19 (-110.1%)	0.35 (-8.8%)	0.02 (60.4%)
	Shape	2.07 (67.7%)	0.24 (93.3%)	0.17 (90.9%)	0.49 (93.0%)	0.11 (84.2%)	0.20 (90.7%)

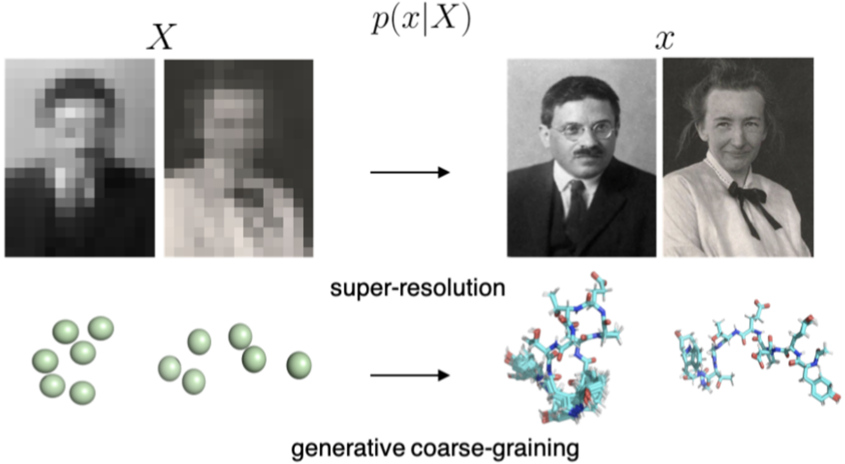
Generative Coarse-Graining of Molecular Conformations

Wujie Wang, Minkai Xu, **Chen Cai**, Benjamin Kurt Miller, Tess Smidt, Yusu Wang, Jian Tang, Rafael Gomez-Bombarelli

Arxiv 2022, under submission

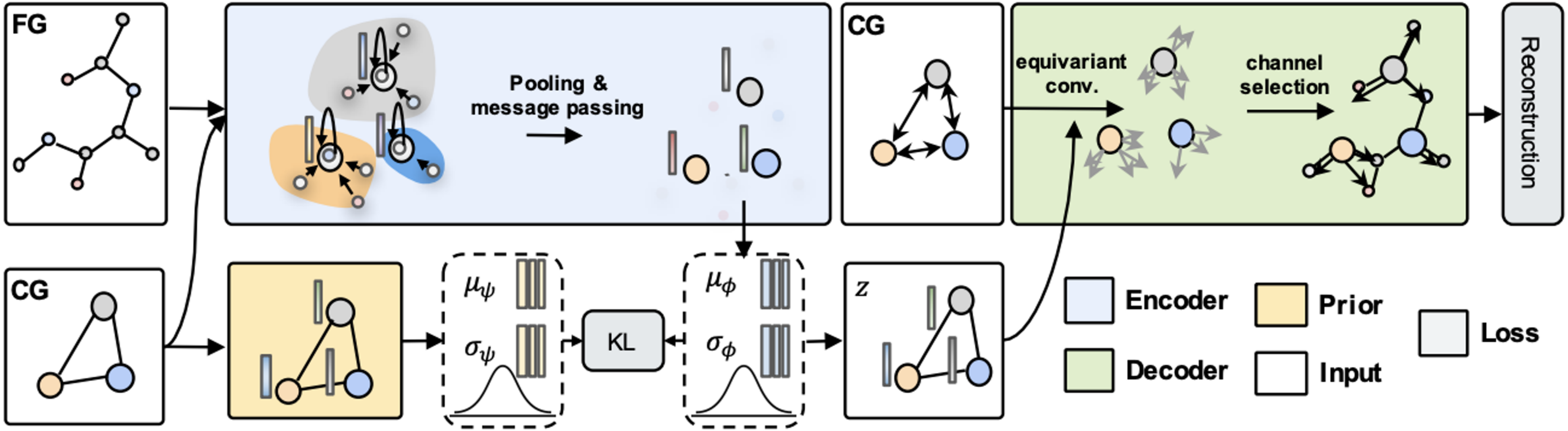
Generative Coarse-Graining of Molecular Conformations

- Coarse-Graining: speed up molecule dynamics (MD) simulation
- Generate novel molecule configurations
- Super resolution for geometric graphs
- Rotation equivariant & handle vector (type 1) features

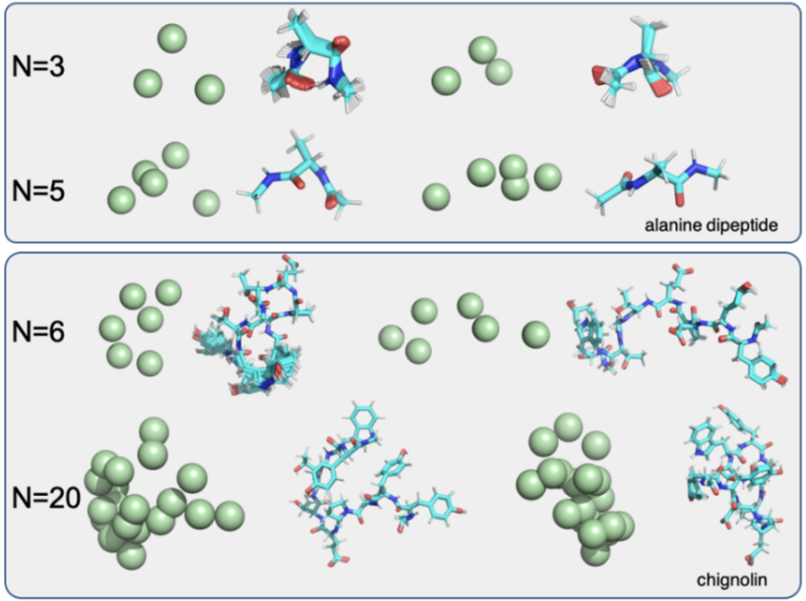


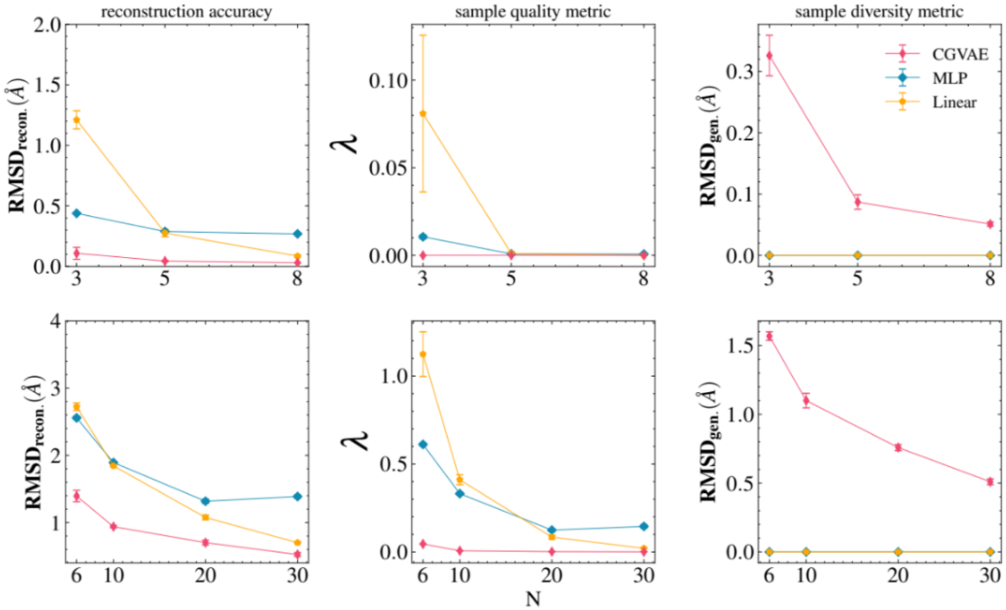
Framework

- Variational autoencoder framework
- Fix coarse graining map
- O(3) equivariant graph encoder & decoder
- Test on 2 systems: alanine dipeptide and chignolin

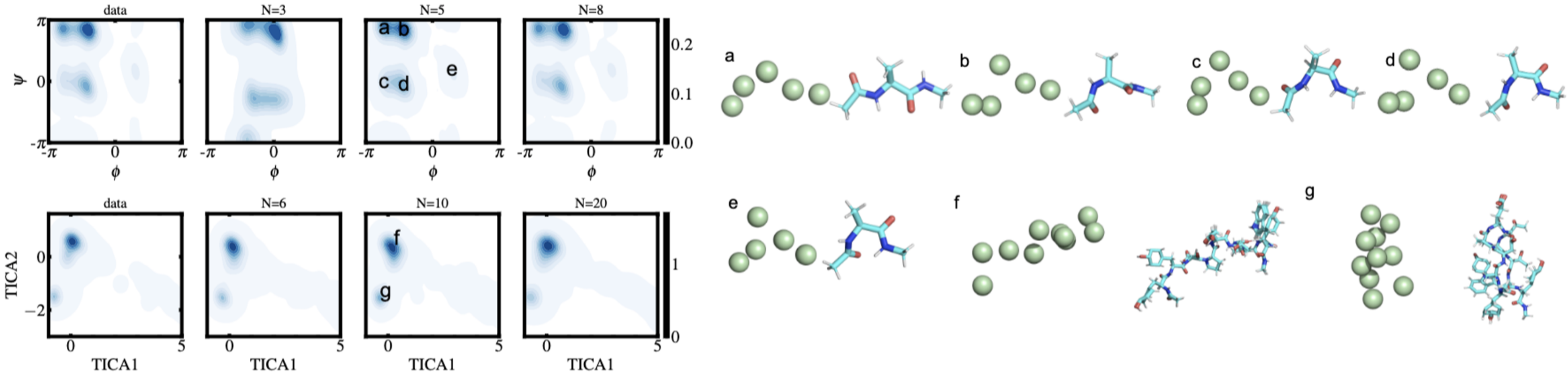


Results


N=3
N=5
alanine dipeptide
N=6
N=20
chignolin


reconstruction accuracy
sample quality metric
sample diversity metric
CGVAE
MLP
Linear
$RMSD_{recon.}$ (Å)
λ
$RMSD_{gen.}$ (Å)
N

Results



Future Directions

- Characterize expressive power of IGN-small
- Can IGN converges after edge smoothing?
- Investigating the convergence of GNN in the manifold setting
- Hardness result of learning combinatorial optimization with GNN

Thank you!