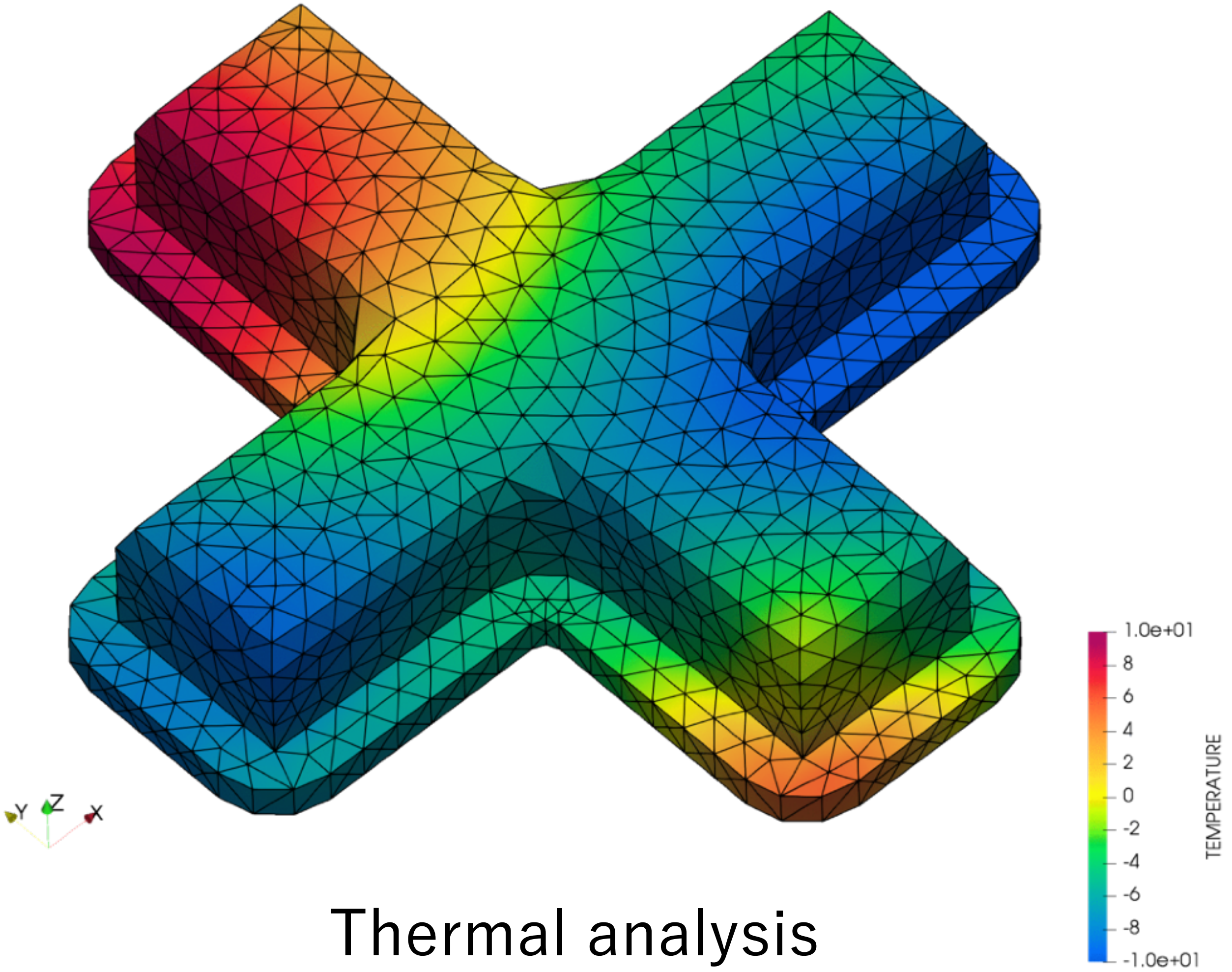
Isometric Transformation Invariant and Equivariant Graph Convolutional Networks

ICLR 2021

Masanobu Horie[1,2] Naoki Morita[1,2] Toshiaki Hishinuma[2]
Yu Ihara[2] Naoto Mitsume[2]

[1] University of Tsukuba
[2] Research Institute for Computational Science Co. Ltd.

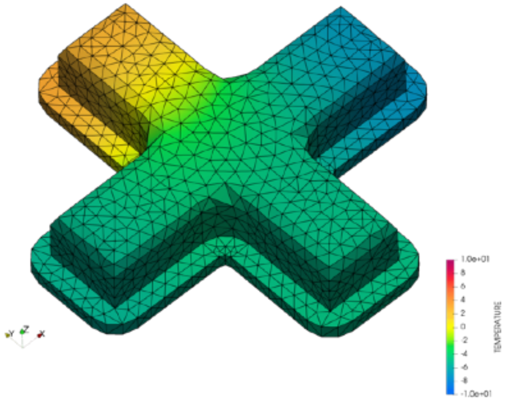
Motivation: Accelerate Physical Simulation



Thermal analysis

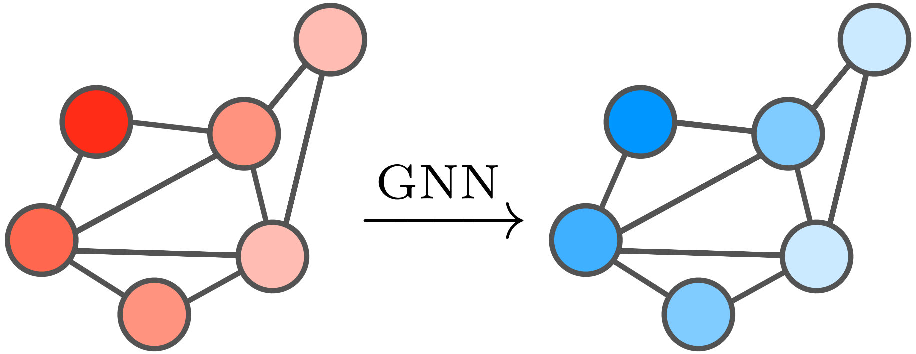
- Physical simulation is essential in manufacturing processes
- Long computation time limits the capacity of manufacturing
- Replace heavy physical simulation with machine learning models

Motivation: Accelerate Physical Simulation

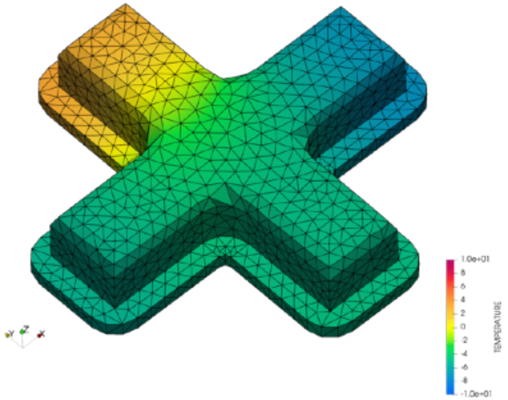


Machine learning for physical simulation requires:

1. **Graph neural networks** (GNN)
   to handle unstructured meshes

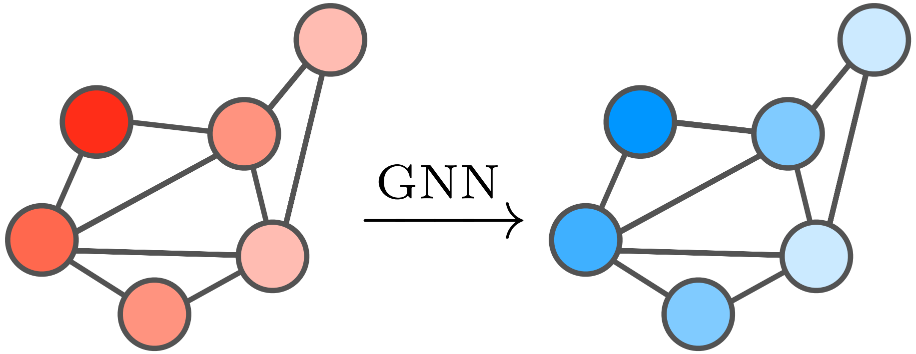


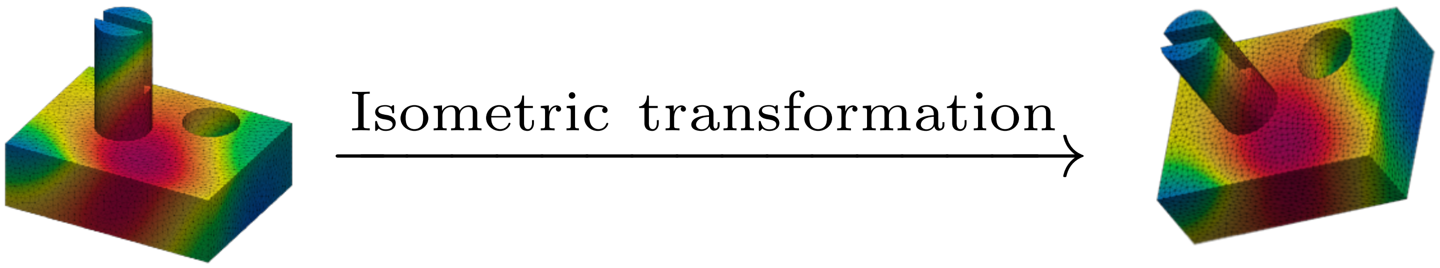
Motivation: Accelerate Physical Simulation



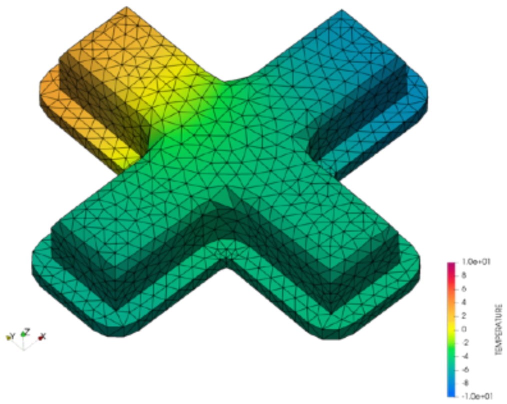
Machine learning for physical simulation requires:

1. **Graph neural networks** (GNN)
   to handle unstructured meshes
2. **Isometric transformation** (E(n)) **equivariance**
   to deal with physical phenomena



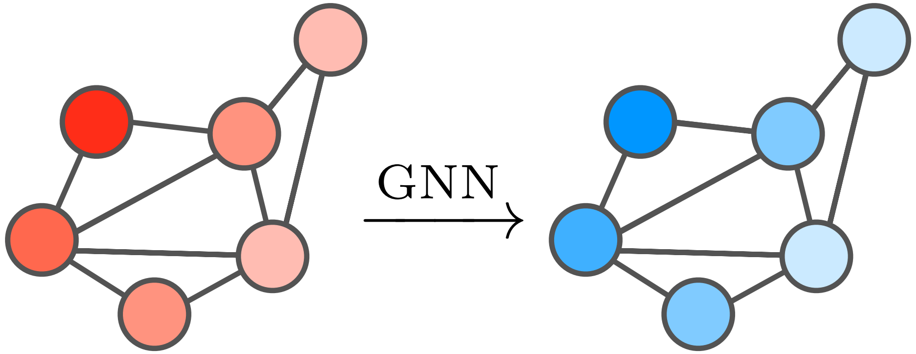


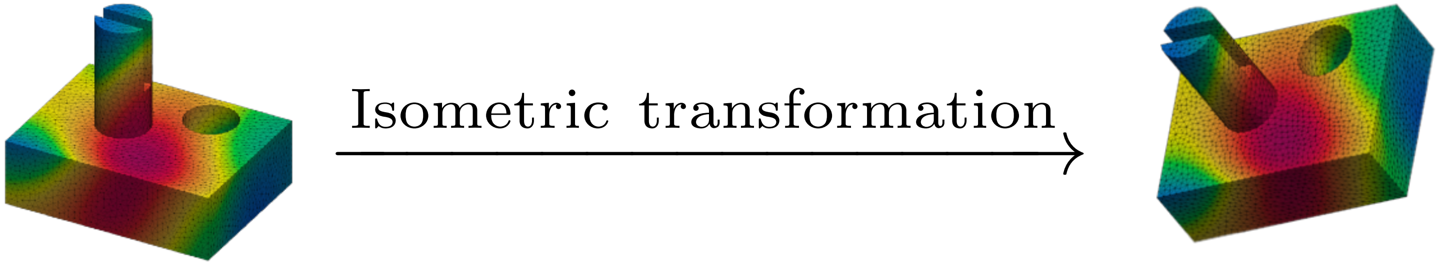
Motivation: Accelerate Physical Simulation



Machine learning for physical simulation requires:

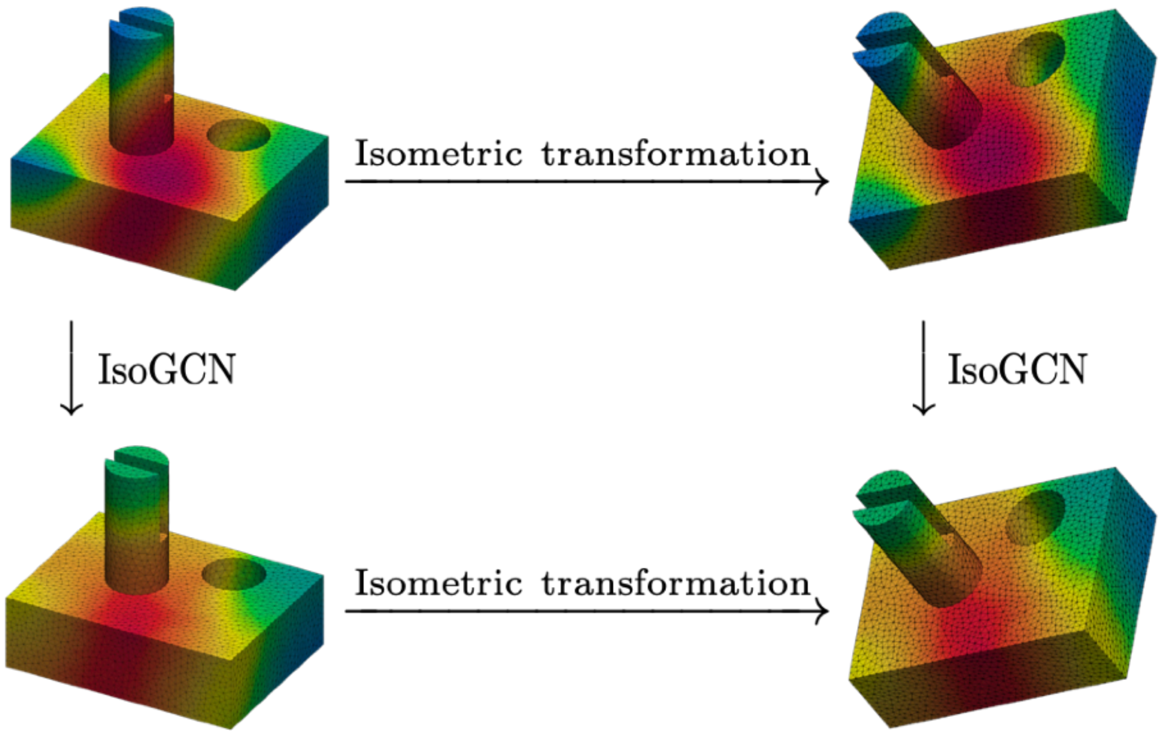
1. **Graph neural networks** (GNN)
   to handle unstructured meshes
2. **Isometric transformation** (E(n)) **equivariance**
   to deal with physical phenomena
3. **Shorter computation time**
   compared to the conventional physical simulation method





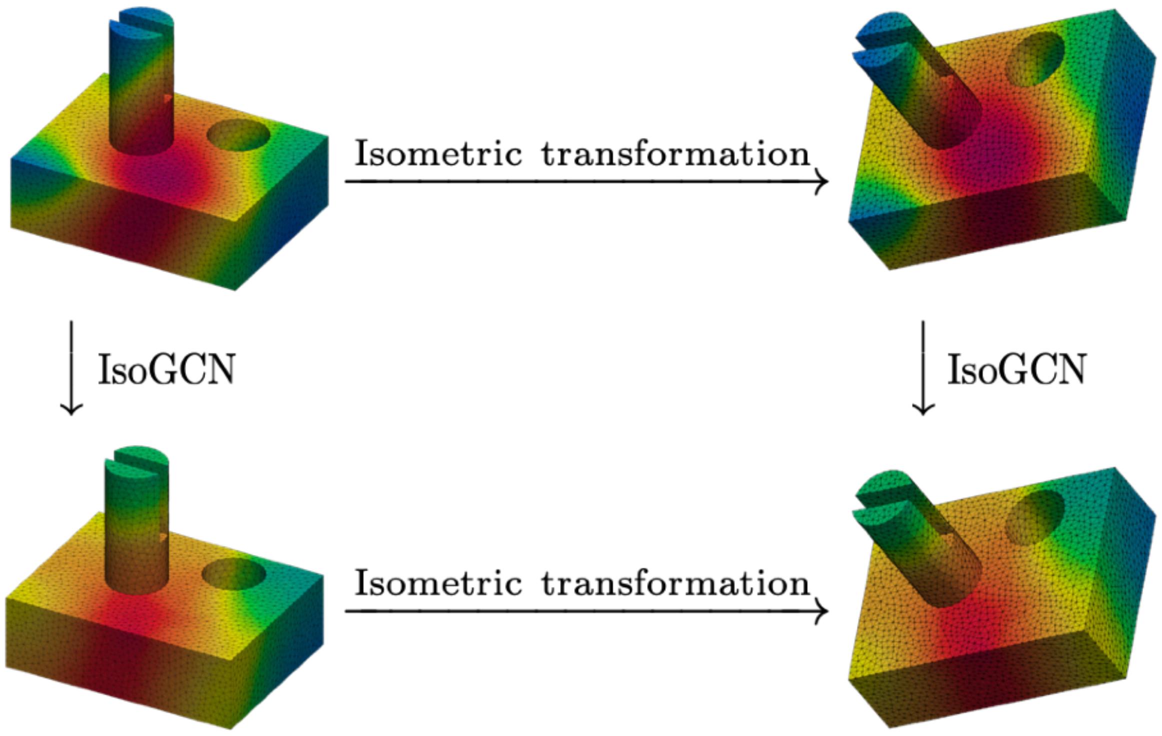
Overview: GCN with Symmetries

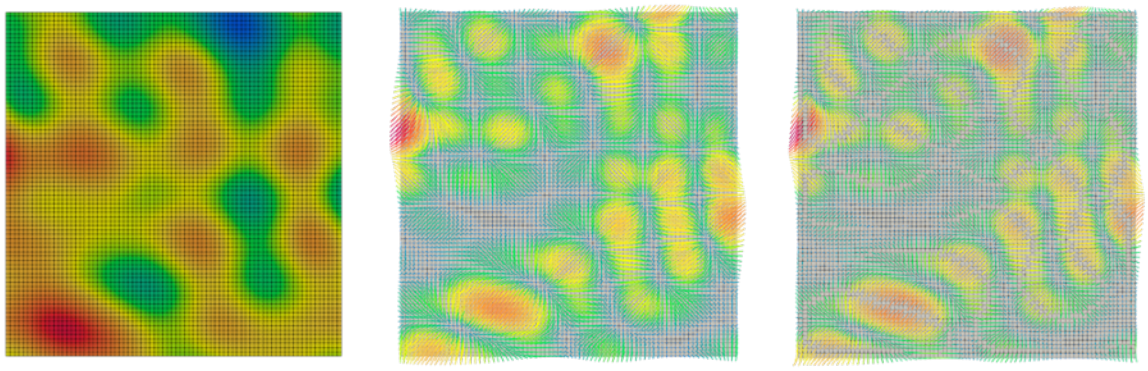
- We developed isometric transformation invariant and equivariant GCNs (IsoGCNs)



Overview: GCN with Symmetries

- We developed isometric transformation invariant and equivariant GCNs (IsoGCNs)
- IsoGCN can approximate differential operators

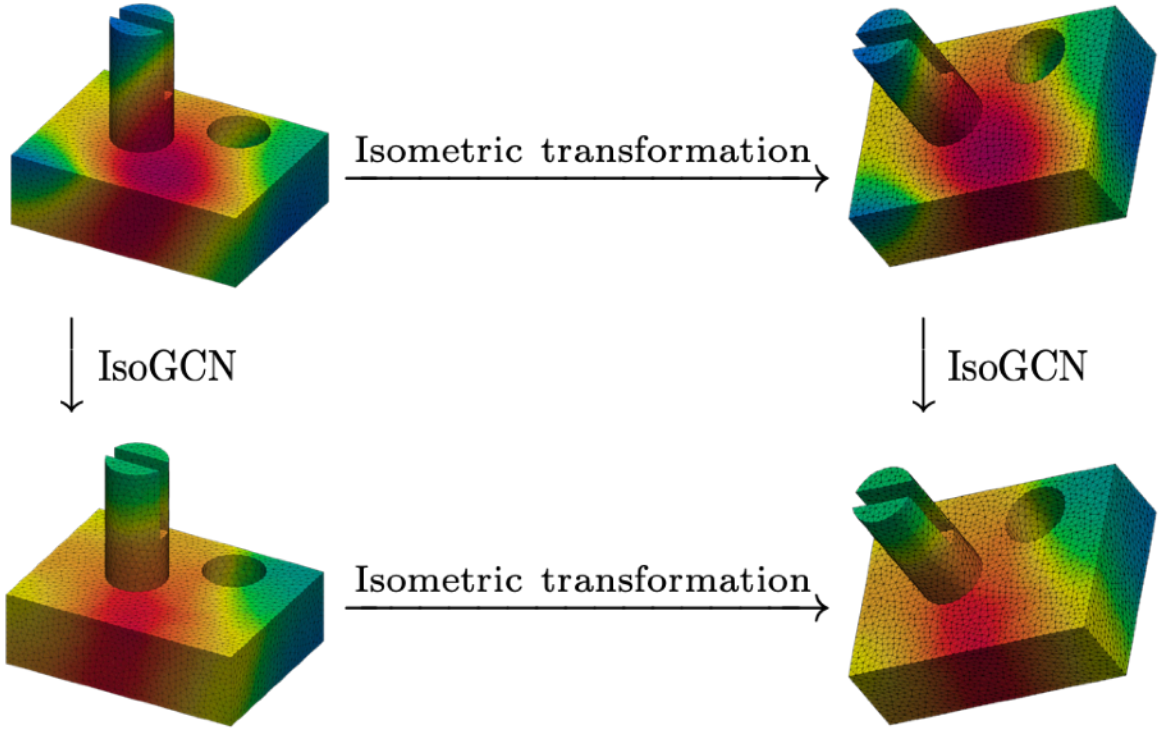


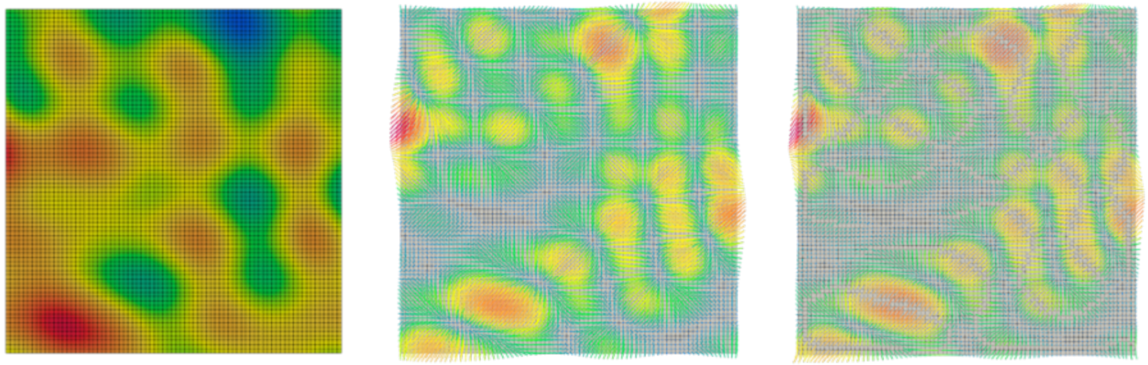


Scalar field (input) Gradient field (ground truth) Gradient field (inference)

Overview: GCN with Symmetries

- We developed isometric transformation invariant and equivariant GCNs (IsoGCNs)
- IsoGCN can approximate differential operators
- IsoGCN can infer simulation results faster than FEA (finite element analysis)



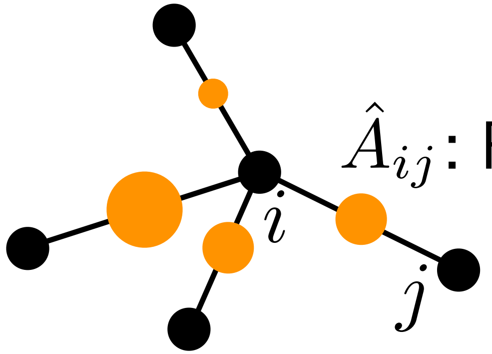


Scalar field (input) Gradient field (ground truth) Gradient field (inference)

Key Idea: GCN with Vector-Valued Adjacency Matrix

- GCN[1]'s message passing

$$H_{\mathrm{out}} = \hat{A} H_{\mathrm{in}} W$$



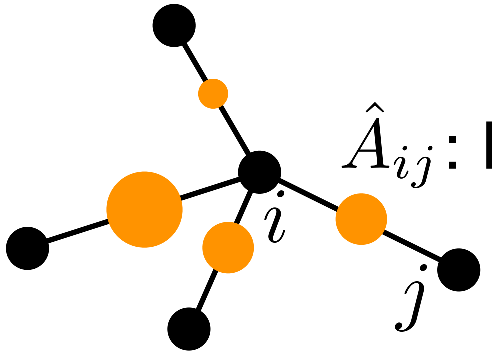
$\hat{A}_{ij}$: Renormalized adjacency matrix

[1]: Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Key Idea: GCN with Vector-Valued Adjacency Matrix

- GCN[1]'s message passing

$$H_{\mathrm{out}} = \hat{A} H_{\mathrm{in}} W$$

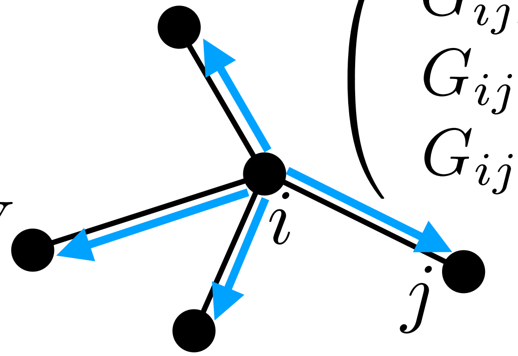


$\hat{A}_{ij}$: Renormalized adjacency matrix

- IsoGCN's message passing:

  - Scalar to vector

$$\begin{pmatrix} H_{\mathrm{out},x} \\ H_{\mathrm{out},y} \\ H_{\mathrm{out},z} \end{pmatrix} = \begin{pmatrix} G_{::,x} \\ G_{::,y} \\ G_{::,z} \end{pmatrix} H_{\mathrm{in}} W$$



$$\begin{pmatrix} G_{ij,x} \\ G_{ij,y} \\ G_{ij,z} \end{pmatrix} = \boldsymbol{x}_j - \boldsymbol{x}_i$$

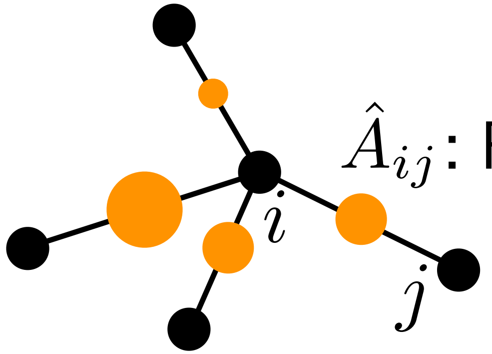
IsoAM : relative vertex positions

[1]: Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Key Idea: GCN with Vector-Valued Adjacency Matrix

- GCN[1]'s message passing

$$H_{\mathrm{out}} = \boxed{\hat{A}} H_{\mathrm{in}} W$$

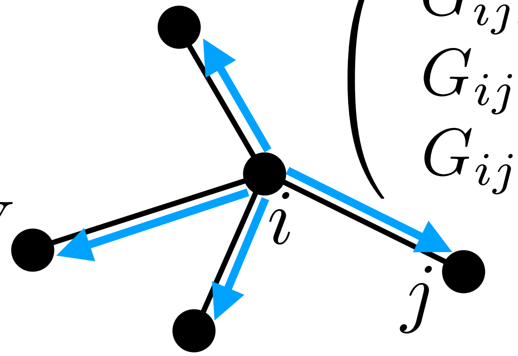


$\hat{A}_{ij}$: Renormalized adjacency matrix

- IsoGCN's message passing:

  - Scalar to vector

$$\begin{pmatrix} H_{\mathrm{out},x} \\ H_{\mathrm{out},y} \\ H_{\mathrm{out},z} \end{pmatrix} = \boxed{\begin{pmatrix} G_{::,x} \\ G_{::,y} \\ G_{::,z} \end{pmatrix}} H_{\mathrm{in}} W$$



$$\begin{pmatrix} G_{ij,x} \\ G_{ij,y} \\ G_{ij,z} \end{pmatrix} = \boldsymbol{x}_j - \boldsymbol{x}_i$$

IsoAM :

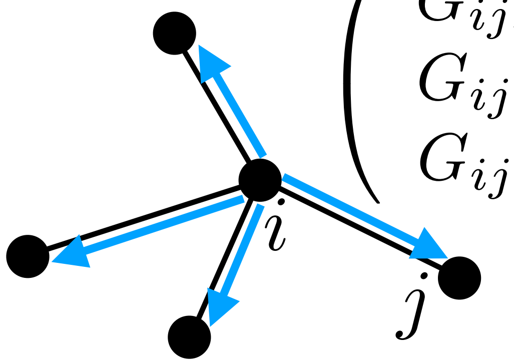
relative vertex positions

[1]: Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Key Idea: GCN with Vector-Valued Adjacency Matrix

- GCN[1]'s message passing

$$H_{\mathrm{out}} = \boxed{\hat{A}} H_{\mathrm{in}} W$$



$$\begin{pmatrix} G_{ij,x} \\ G_{ij,y} \\ G_{ij,z} \end{pmatrix} = \boldsymbol{x}_j - \boldsymbol{x}_i$$

IsoAM :

relative vertex positions

- IsoGCN's message passing:

  - Scalar to vector

$$\begin{pmatrix} H_{\mathrm{out},x} \\ H_{\mathrm{out},y} \\ H_{\mathrm{out},z} \end{pmatrix} = \boxed{\begin{pmatrix} G_{::,x} \\ G_{::,y} \\ G_{::,z} \end{pmatrix}} H_{\mathrm{in}} W$$

- IsoGCN incorporates graphs' geometry information by simply replacing the adjacency matrices with IsoAMs

[1]: Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Key Idea: GCN with Vector-Valued Adjacency Matrix

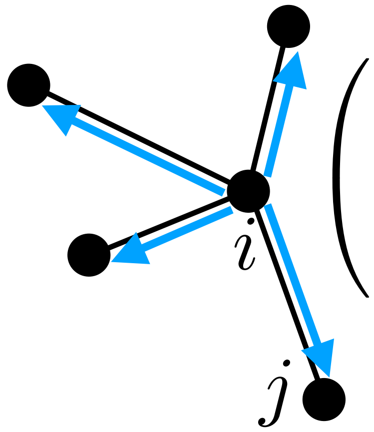
- GCN[1]'s message passing

$$H_{\mathrm{out}} = \boxed{\hat{A}} H_{\mathrm{in}} W$$

- IsoGCN's message passing:

  - Scalar to vector

$$\begin{pmatrix} H_{\mathrm{out},x} \\ H_{\mathrm{out},y} \\ H_{\mathrm{out},z} \end{pmatrix} = \boxed{\begin{pmatrix} G_{::,x} \\ G_{::,y} \\ G_{::,z} \end{pmatrix}} H_{\mathrm{in}} W$$



$$\begin{pmatrix} G_{ij,x} \\ G_{ij,y} \\ G_{ij,z} \end{pmatrix} = \boldsymbol{x}_j - \boldsymbol{x}_i$$

IsoAM :
relative vertex positions

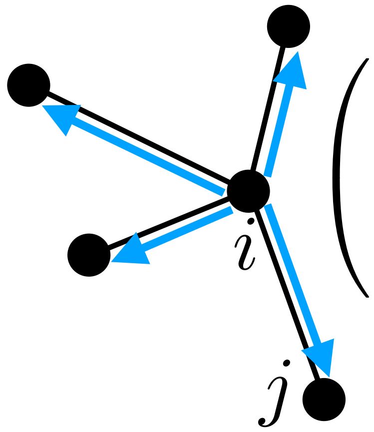
- IsoGCN incorporates graphs' geometry information by simply replacing the adjacency matrices with IsoAMs
- IsoAM is equivariant

[1]: Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Key Idea: GCN with Vector-Valued Adjacency Matrix

- IsoGCN's message passing:
  - Scalar to vector

$$\begin{pmatrix} H_{\mathrm{out},x} \\ H_{\mathrm{out},y} \\ H_{\mathrm{out},z} \end{pmatrix} = \begin{pmatrix} G_{::,x} \\ G_{::,y} \\ G_{::,z} \end{pmatrix} H_{\mathrm{in}} W$$



$$\begin{pmatrix} G_{ij,x} \\ G_{ij,y} \\ G_{ij,z} \end{pmatrix} = \boldsymbol{x}_j - \boldsymbol{x}_i$$

IsoAM :
relative vertex positions

  - Vector to scalar

$$H_{\mathrm{out}} = \begin{pmatrix} G_{::,x} \\ G_{::,y} \\ G_{::,z} \end{pmatrix} \cdot \begin{pmatrix} H_{\mathrm{in},x} \\ H_{\mathrm{in},y} \\ H_{\mathrm{in},z} \end{pmatrix} W$$

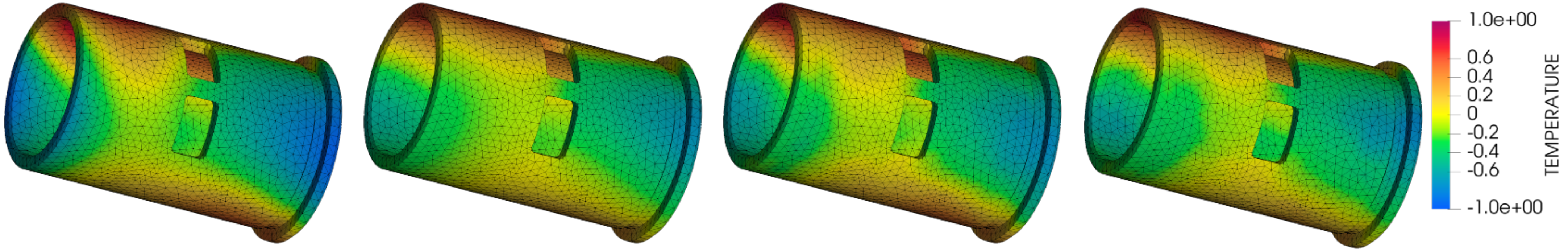
- IsoGCN can convert any rank tensors to any rank tensors with keeping equivariance

  - Scalar to rank-2 tensor

$$\begin{pmatrix} H_{\mathrm{out},xx} & H_{\mathrm{out},xy} & H_{\mathrm{out},xz} \\ H_{\mathrm{out},yx} & H_{\mathrm{out},yy} & H_{\mathrm{out},yz} \\ H_{\mathrm{out},zx} & H_{\mathrm{out},zy} & H_{\mathrm{out},zz} \end{pmatrix} = \begin{pmatrix} G_{::,x}G_{::,x} & G_{::,x}G_{::,y} & G_{::,x}G_{::,z} \\ G_{::,y}G_{::,x} & G_{::,y}G_{::,y} & G_{::,y}G_{::,z} \\ G_{::,z}G_{::,x} & G_{::,z}G_{::,y} & G_{::,z}G_{::,z} \end{pmatrix} H_{\mathrm{in}} W$$

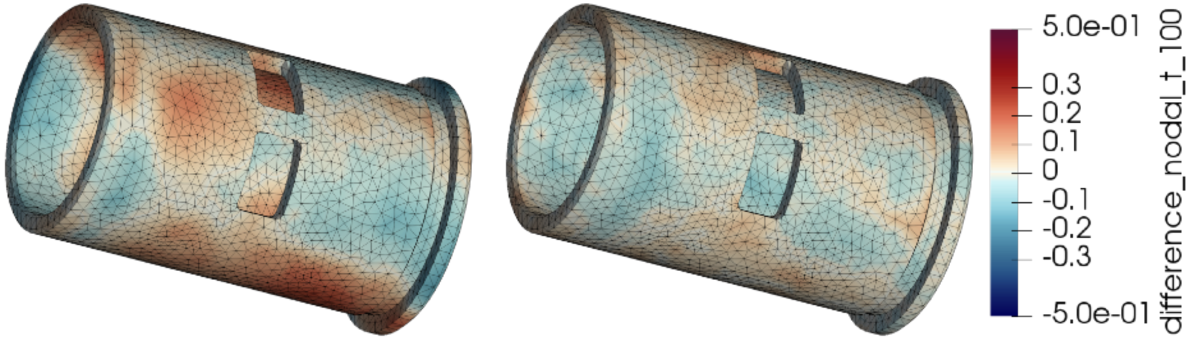
Experiments: Heat Equation Dataset

Initial condition | FEA results (ground truth) | SE(3)-Trans.[2] | IsoGCN (Ours)



- IsoGCN outperformed baseline models
- Baseline equivariant models had to be small to fit on GPU memory (32 GiB)

SE(3)-Trans. (error) | IsoGCN (error)

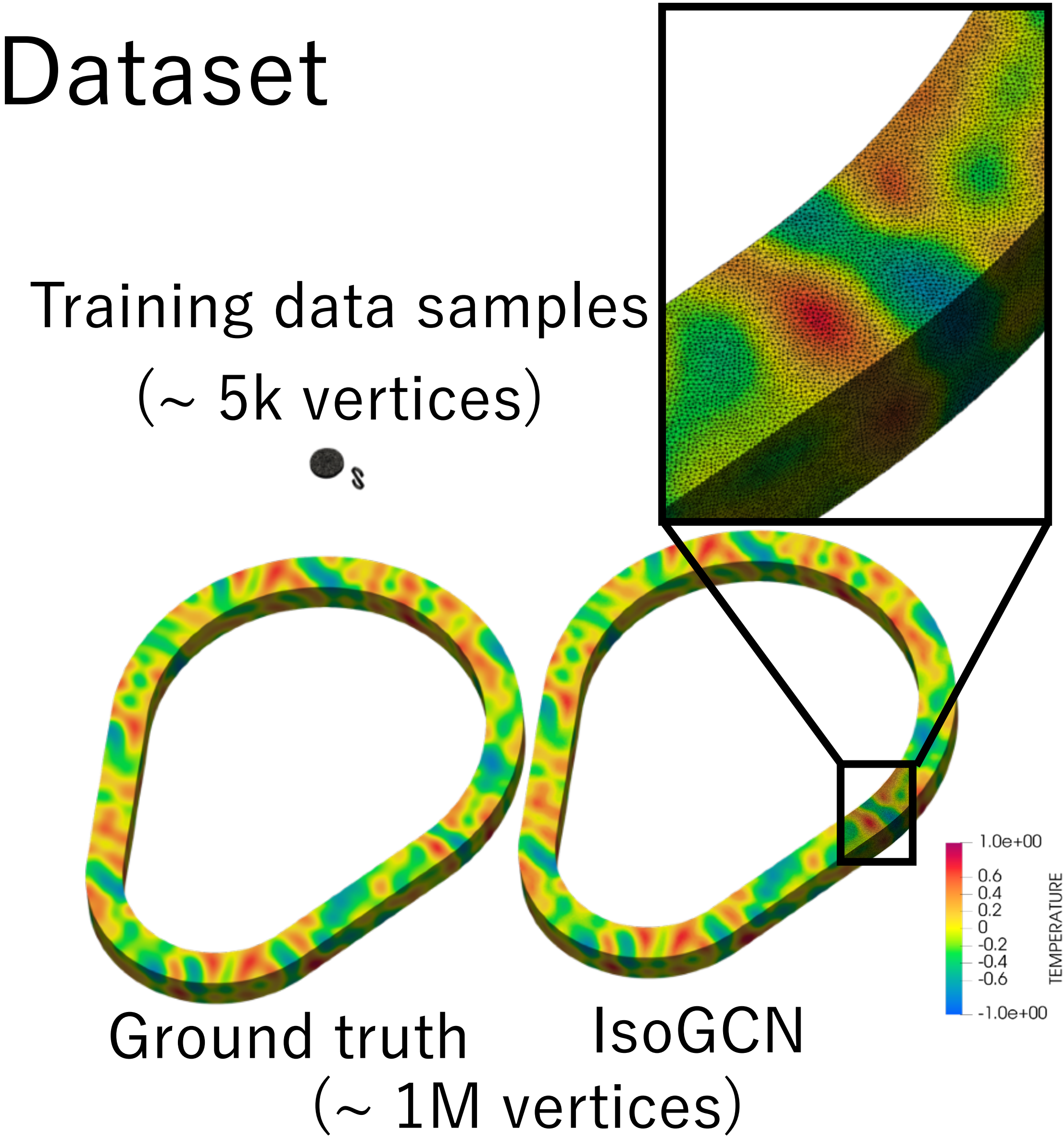


[2]: Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d rototranslation equivariant attention networks. In *Advances in Neural Information Processing Systems*, 33, 2020.

Experiments: Heat Equation Dataset

- IsoGCN can infer faster than the conventional FEA simulation
- IsoGCN can scale more than 1M vertices while the baseline equivariant models cannot

	155,019 vertices		**1,011,301 vertices**	
	Loss (10^{-4})	**Time** [s]	**Loss** (10^{-4})	**Time** [s]
FEA ($\Delta t = 1.0$)	6.1	181.7	2.9	1656.5
FEA ($\Delta t = 0.5$)	0.4	288.0	0.2	2884.2
TFN[3]	30.1	400.9	OOM	OOM
SE(3)-Transformer	80.3	271.1	OOM	OOM
IsoGCN (Ours)	4.9	**84.1**	3.9	**648.4**



[3]: Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint* arXiv:1802.08219, 2018.

Conclusion

- IsoGCN is the fast and scalable equivariant graph neural network
- Suitable to learn physical simulation because of the scalability and equivariance
- Plan to apply IsoGCN to various physical phenomena