CSC2547 3D & Geometric Deep Learning

Isometric Transformation Invariant and Equivariant Graph Convolutional Networks

Masanobu Horie, Naoki Morita, Toshiaki Hishinuma, Yu Ihara, Naoto Mitsume

March 23rd, 2021

Presenter: Sejin Kim

Instructor: Animesh Garg

UNIVERSITY OF TORONTO

Motivation and Main Problem

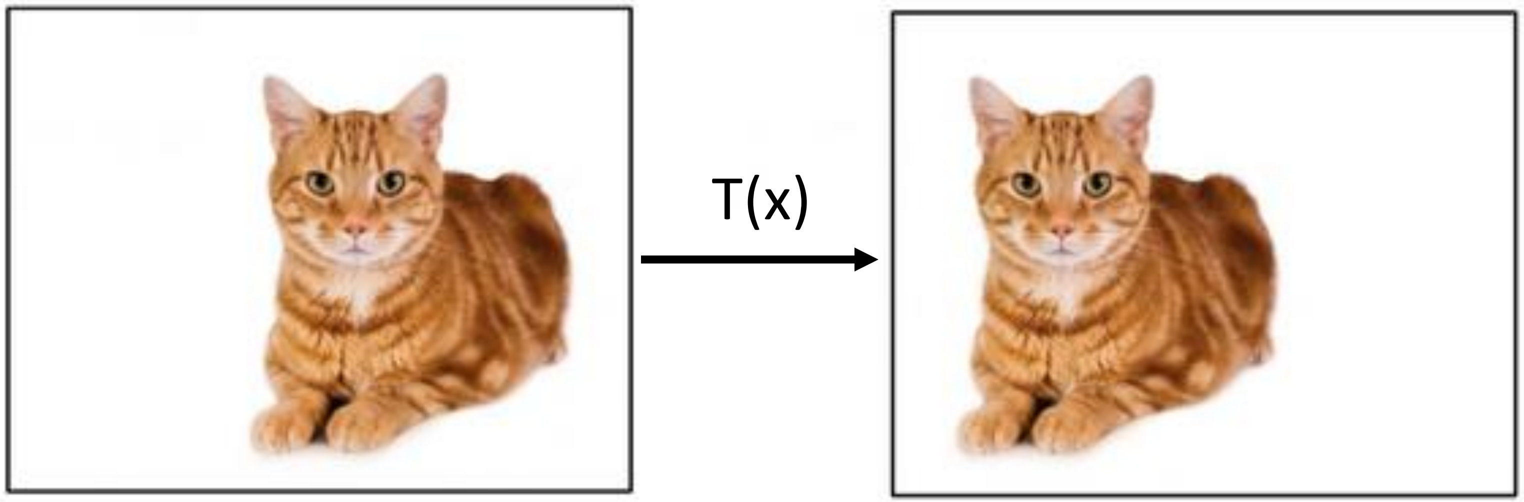
- Isometric transformations (IsoTr)

Translation Invariant

f(x) = Cat

- Classification: $f(x) = R^d \rightarrow R$
- Transformation: $T(x) = R^d \rightarrow R^d$

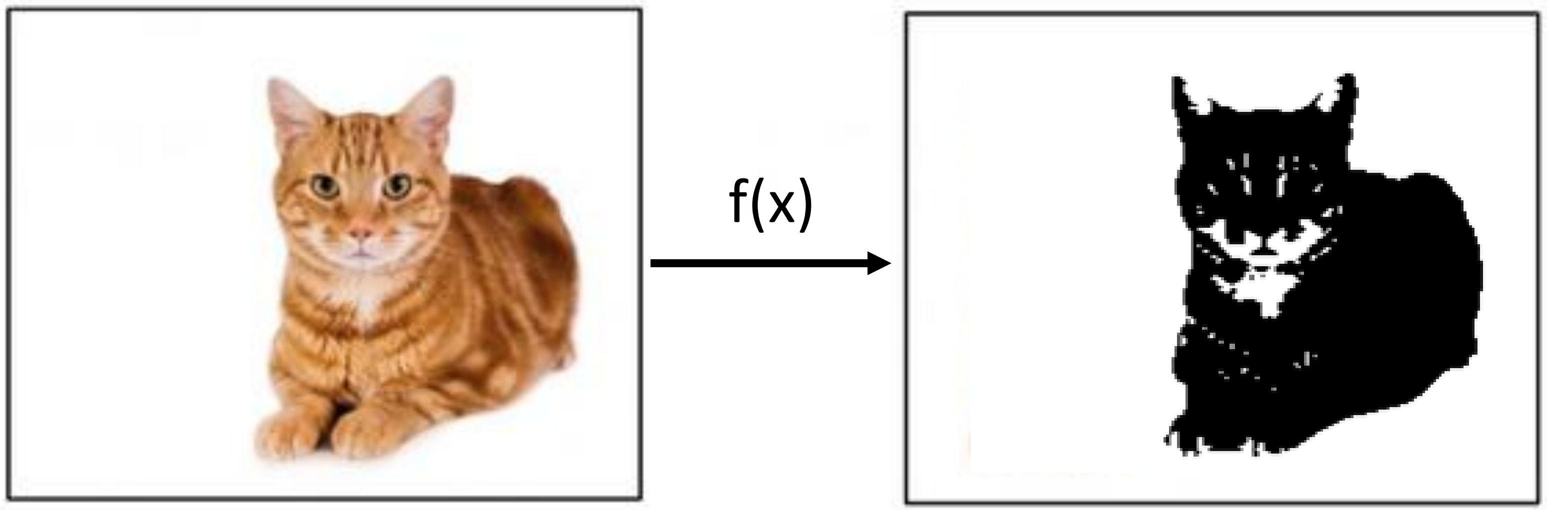
Translation Invariant



$f(x) = $ Cat $\qquad$ $f(x) = $ Cat

- Translation Invariant: $\quad f(x) = f(T(x))$

Translation Equivariant



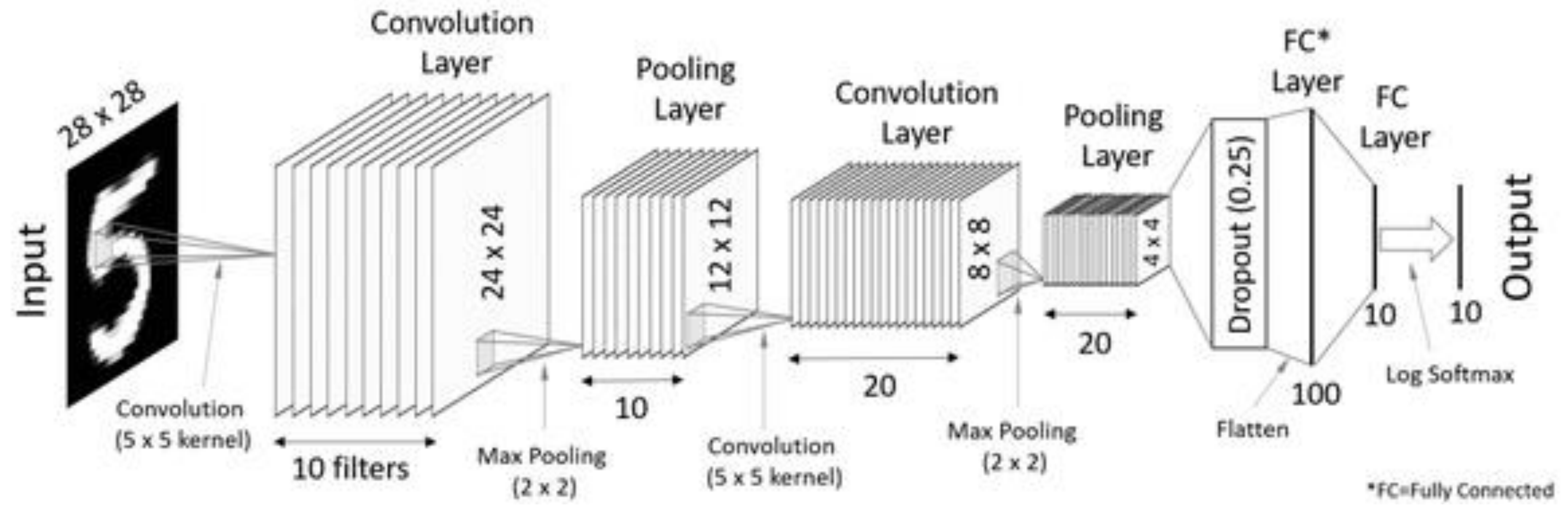
- Segmentation: $f(x) = R^d \rightarrow R^d$
- Transformation: $T(x) = R^d \rightarrow R^d$

Translation Equivariant

$\downarrow T(x)$

- Translation equivariant: $T(f(x)) = f(T(x))$

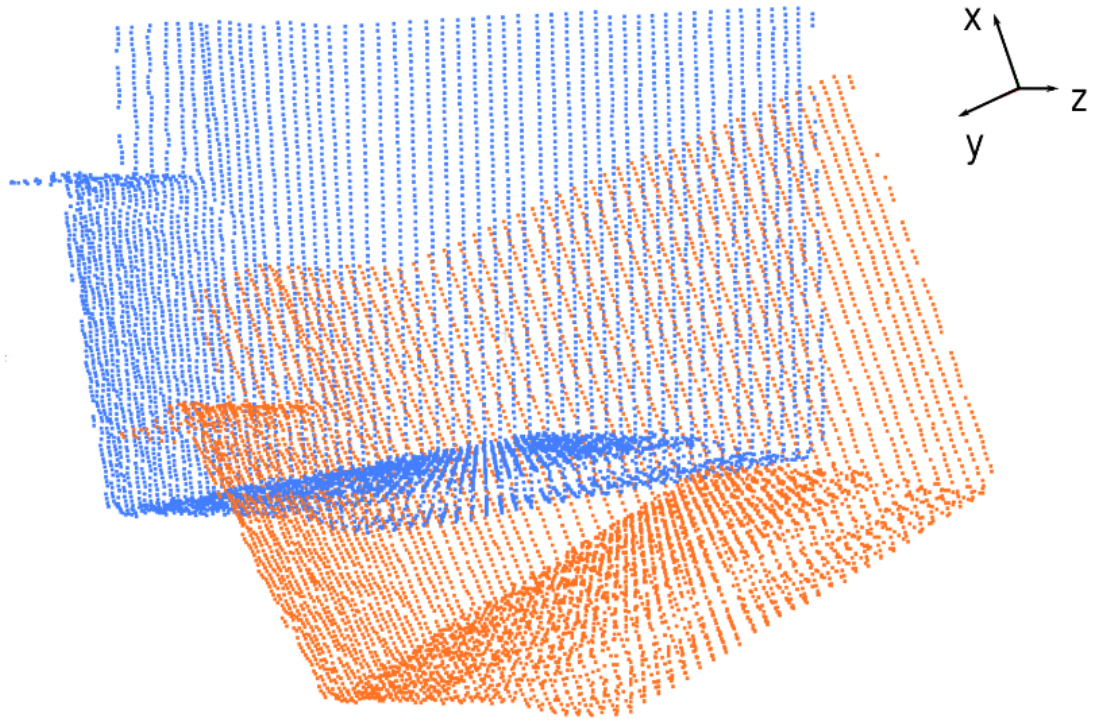
Invariance vs Equivariance

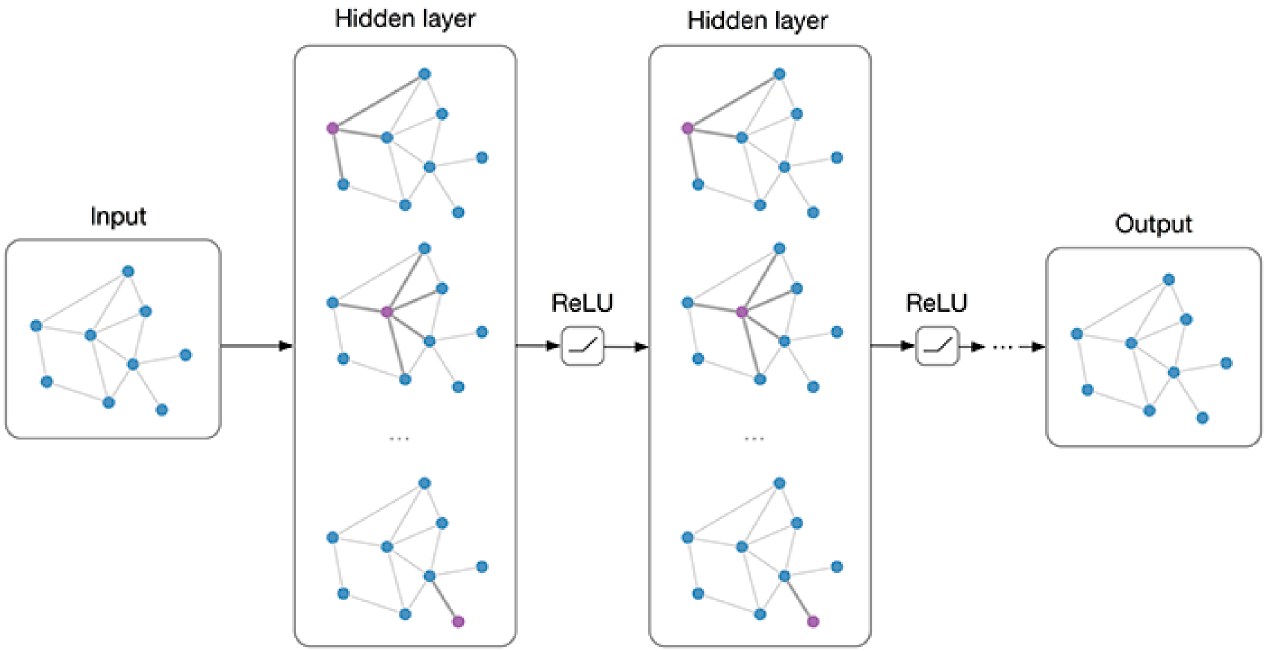


Translation **Equivariant**

Translation **Invariant**

Non-Euclidean Data



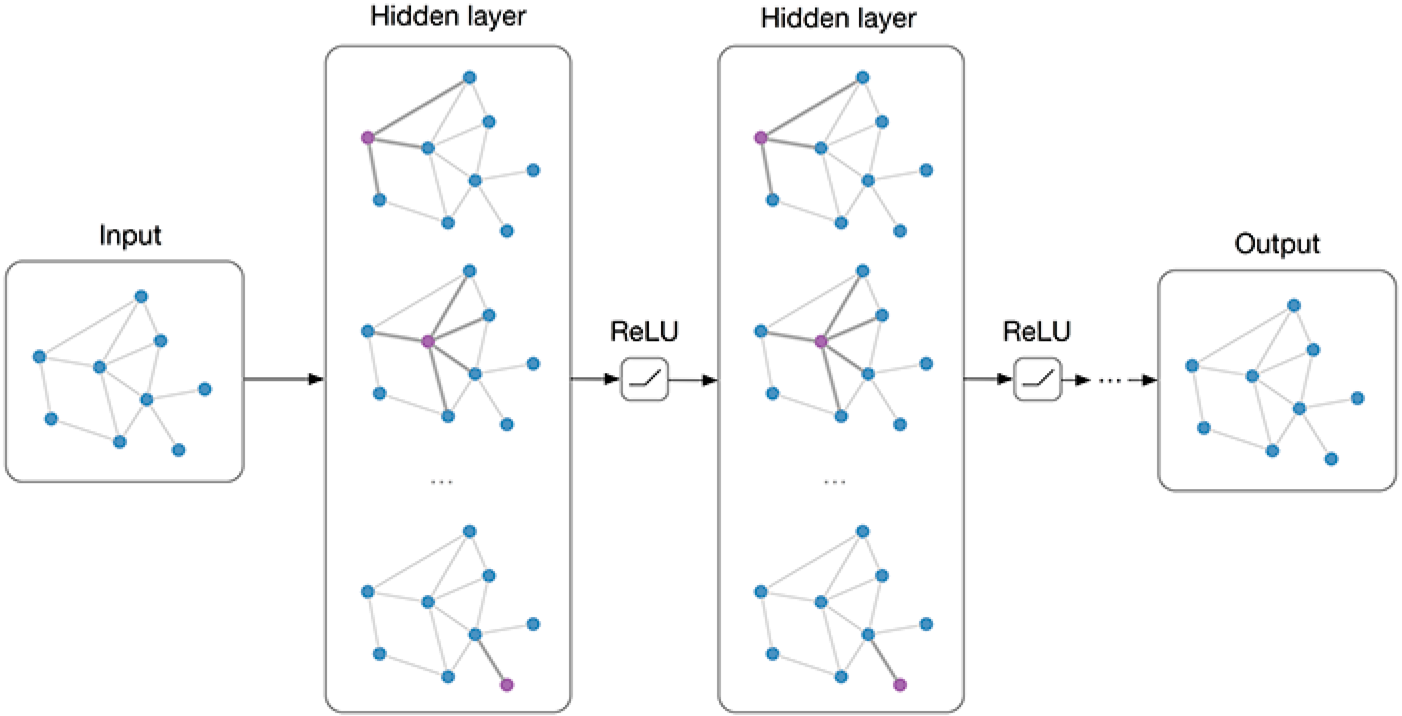


Contributions

- *IsoGCN*: Construct IsoTr **invariant + equivariant** GCNs
  - Introduced novel adjacency matrix, *IsoAM*
- Demonstrate competitive performance in tasks related to **physical simulations**
- High scalability: **up to 1M vertices** and achieve inference faster than conventional element analysis

Graph Convolutional Networks

- Simplification of message-passing GNN
- Increased computational efficiency



Graph Convolutional Network

$$\boldsymbol{H}_{\text{out}} = \sigma(\hat{\boldsymbol{A}}\boldsymbol{H}_{\text{in}}\boldsymbol{W})$$

• H_{out}: Hidden Layer output	$\lvert V \rvert$ x f_{out}
• H_{in}: Hidden Layer input	$\lvert V \rvert$ x f_{in}
• A: Adjacency matrix	$\lvert V \rvert$ x $\lvert V \rvert$
• W: Trainable weights	f_{in} x f_{out}

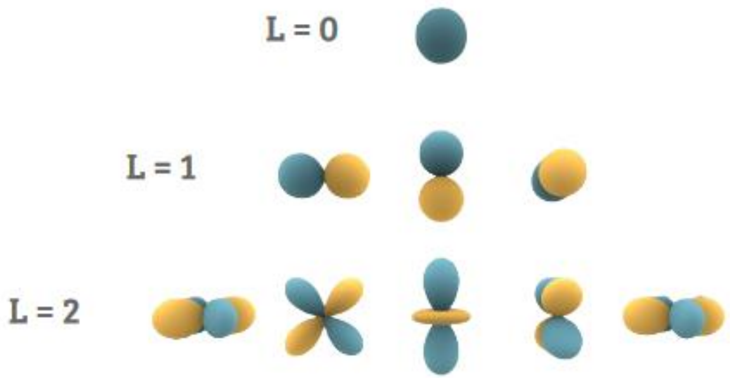
- $|V|$ = number of vertices

Previous Works

- CNN-based
  - Group Equivariant CNNs (Cohen and Welling, 2016)
  - 3D Steerable CNNs (Weiler et al, 2018)

- Tensor Field Network (Thomas et al, 2018):
  - Rotation + translation equivariant NN **for point clouds**
  - Using 3D spherical harmonics

- SE(3) Transformer (2020): TFN w/ self-attention



Tensor Field Network

$$\boldsymbol{H}_{\text{out}} = \sigma(\hat{\boldsymbol{A}}\boldsymbol{H}_{\text{in}}\boldsymbol{W})$$

$$\tilde{\mathbf{H}}^{(l)}_{\text{out},i} = w^{ll}\,\tilde{\mathbf{H}}^{(l)}_{\text{in},i} + \sum_{k\geq 0}\sum_{j\neq i} \boldsymbol{W}^{lk}(\boldsymbol{x}_j - \boldsymbol{x}_i)\,\tilde{\mathbf{H}}^{(k)}_{\text{in},j},$$

$$\boldsymbol{W}^{lk}(\boldsymbol{x}) = \sum_{J=|k-l|}^{k+l} \phi^{lk}_J(\|\boldsymbol{x}\|) \sum_{m=-J}^{J} Y_{Jm}(\boldsymbol{x}/\|\boldsymbol{x}\|)\boldsymbol{Q}^{lk}_{Jm},$$

- ϕ: trainable function
- Y_Jm: mth component of Jth spherical harmonics
- Q: Clebsch-Cordan coefficient

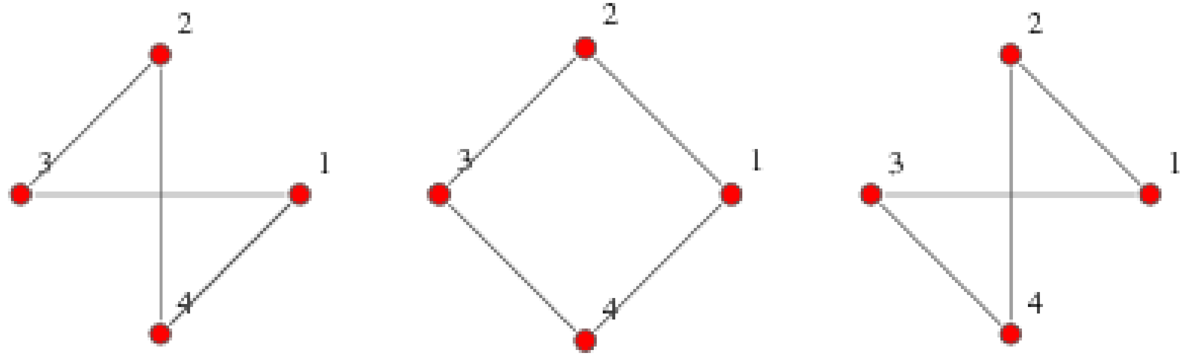
Tensor Field Network

$$\tilde{\mathbf{H}}^{(l)}_{\text{out},i} = w^{ll}\,\tilde{\mathbf{H}}^{(l)}_{\text{in},i} + \sum_{k\geq 0}\sum_{j\neq i} \boldsymbol{W}^{lk}(\boldsymbol{x}_j - \boldsymbol{x}_i)\,\tilde{\mathbf{H}}^{(k)}_{\text{in},j},$$

$$\boldsymbol{W}^{lk}(\boldsymbol{x}) = \sum_{J=|k-l|}^{k+l} \phi^{lk}_J(\|\boldsymbol{x}\|) \sum_{m=-J}^{J} Y_{Jm}(\boldsymbol{x}/\|\boldsymbol{x}\|) Q^{lk}_{Jm},$$

- ϕ: trainable function
- Y_Jm: m^{th} component of J^{th} spherical harmonics
- Q: Clebsch-Cordan coefficient

Adjacency Matrix

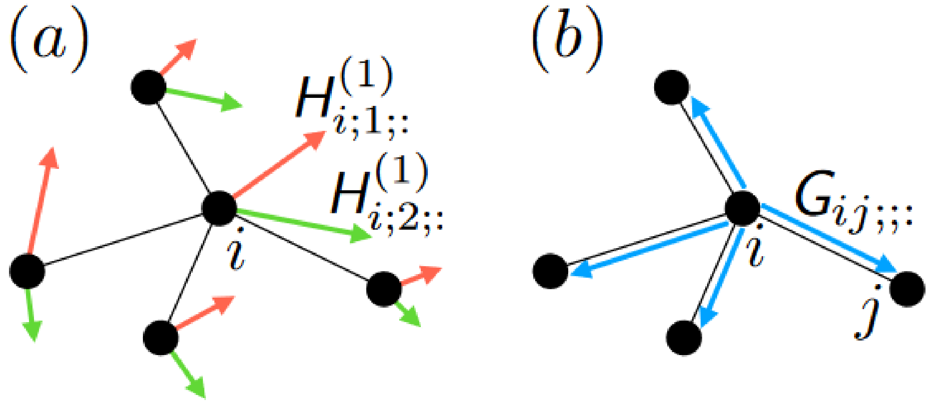


$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

- |V| x |V|
- 1 between connected vertices
  - Weighted AM can have varying values
- 0 for disconnected

IsoAM $\mathbf{G} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|\times d}$

$$\mathbb{R}^d \ni \mathbf{G}_{ij;;:} := \sum_{k,l\in\mathcal{V},k\neq l} \boldsymbol{T}_{ijkl}(\boldsymbol{x}_k - \boldsymbol{x}_l)$$



$$\begin{aligned}\mathbf{G}_{ij;;:} &= \delta_{il}\delta_{jk}A_{ij}\boldsymbol{I}(\boldsymbol{x}_k - \boldsymbol{x}_l) \\ &= A_{ij}(\boldsymbol{x}_j - \boldsymbol{x}_i)\end{aligned}$$

- d: Euclidean dims
- G_ij (d): slice in spatial index of G
- T_ijkl (d x d): **untrainable** transformation invariant + orthogonal transformation equivariant rank-2 tensor

Proposition 1

- T: x ↦ Ux + t
  - t: translation
  - U: orthogonal transformation

$$T : G_{ij;;k} \mapsto \sum_l U_{kl} G_{ij;;l}.$$

Isometrically Transformed IsoAM ↦ Orthogonally Equivariant
(+ Translational Invariant)

3 Operations

- Convolution
  - $R^{|V| \times f}, R^{|V| \times |V| \times d} \rightarrow R^{|V| \times f \times d}$

$$(\mathbf{G} * \mathbf{H}^{(0)})_{i;g;k} := \sum_j \mathbf{G}_{ij;;k} H^{(0)}_{j;g;}.$$

- Contraction
  - $R^{|V| \times f \times d}, R^{|V| \times |V| \times d} \rightarrow R^{|V| \times f}$

$$(\mathbf{G} \odot \mathbf{H}^{(1)})_{i;g;} := \sum_{j,k} G_{ij;;k} H^{(1)}_{j;g;k}$$

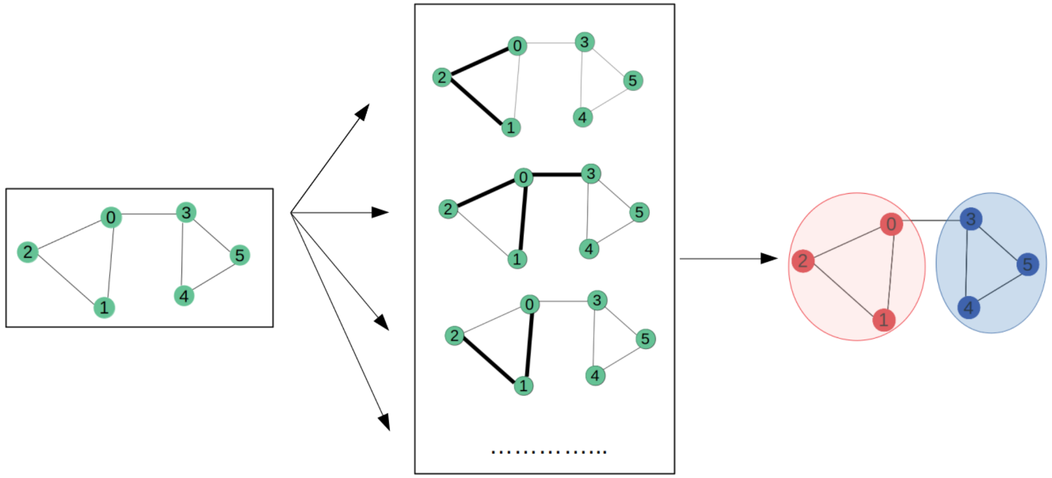
- Tensor Product
  - $R^{|V| \times f \times d^p}, R^{|V| \times |V| \times d} \rightarrow R^{|V| \times f \times d^{(1+p)}}$

$$(\mathbf{G} \otimes \mathbf{H}^{(p)})_{i;g;km_1 m_2 \ldots m_p} := \sum_j \mathbf{G}_{ij;;k} H^{(p)}_{j;g;m_1 m_2 \ldots m_p}$$

Convolution

$$(\mathbf{G} * \mathbf{H}^{(0)})_{i;g;k} := \sum_{j} \mathbf{G}_{ij;;k} H^{(0)}_{j;g;}.$$

- $R^{|V| \times f}, R^{|V| \times |V| \times d} \rightarrow R^{|V| \times f \times d}$



$$\underset{\text{Input}}{\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \end{bmatrix}} * \underset{\text{Kernel}}{\begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}} = \underset{\text{Output}}{\begin{bmatrix} 19 & 25 \\ 37 & 43 \end{bmatrix}}$$

Contraction

$$(\mathbf{G} \odot \mathbf{H}^{(1)})_{i;g;} := \sum_{j,k} G_{ij;;k} H^{(1)}_{j;g;k}$$

- $R^{|V| \times f \times d}$, $R^{|V| \times |V| \times d} \rightarrow R^{|V| \times f}$

$$\begin{bmatrix} A_x & A_y & A_z \end{bmatrix} \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} = A_x B_x + A_y B_y + A_z B_z$$

Tensor Product

$$(\mathbf{G} \otimes \mathbf{H}^{(p)})_{i;g;km_1m_2\ldots m_p} := \sum_j \mathbf{G}_{ij;;k} H^{(p)}_{j;g;m_1m_2\ldots m_p}$$

- $R^{|V| \times f \times d\hat{}p}$, $R^{|V| \times |V| \times d} \rightarrow R^{|V| \times f \times d\hat{}(1+p)}$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 1\begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 2\begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \\ 3\begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 4\begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1\times 0 & 1\times 5 & 2\times 0 & 2\times 5 \\ 1\times 6 & 1\times 7 & 2\times 6 & 2\times 7 \\ 3\times 0 & 3\times 5 & 4\times 0 & 4\times 5 \\ 3\times 6 & 3\times 7 & 4\times 6 & 4\times 7 \end{bmatrix}$$

Proposition 2

$$T : \mathbf{G}_{ij;;k} \mapsto \sum_l U_{kl} \mathbf{G}_{ij;;l}.$$

Proposition 3.2. *The contraction of IsoAMs* $\mathbf{G} \odot \mathbf{G}$ *is isometric transformation invariant, i.e., for any isometric transformation* $\forall \boldsymbol{t} \in \mathbb{R}^3, \boldsymbol{U} \in \mathrm{O}(d), T : \boldsymbol{x} \mapsto \boldsymbol{U}\boldsymbol{x} + \boldsymbol{t}$, $\mathbf{G} \odot \mathbf{G} \mapsto \mathbf{G} \odot \mathbf{G}$.

- $T\colon\ G \odot G \longmapsto \sum U_{km} G_{ij;;m} U_{kn} G_{jl;;n}$

Proposition 2

$\mathbf{G} \odot \mathbf{G}$ *is isometric transformation invariant,*

$$\forall t \in \mathbb{R}^3, U \in \mathrm{O}(d), T : x \mapsto Ux + t, \mathbf{G} \odot \mathbf{G} \mapsto \mathbf{G} \odot \mathbf{G}$$

T(x)

$\mathbf{G} \odot \mathbf{G}$ f(x) = Cat

$\mathbf{G} \odot \mathbf{G}$ f(x) = Cat

Proposition 3

Proposition 3.3. *The tensor product of the IsoAMs* $\mathbf{G} \otimes \mathbf{G}$ *is isometric transformation equivariant in terms of the rank-2 tensor, i.e., for any isometric transformation* $\forall \boldsymbol{t} \in \mathbb{R}^3, \boldsymbol{U} \in \mathrm{O}(d), T : \boldsymbol{x} \mapsto \boldsymbol{U}\boldsymbol{x} + \boldsymbol{t}$*, and* $\forall i, j \in 1, \ldots, |\mathcal{V}|$, $(\mathbf{G} \otimes \mathbf{G})_{ij;;k_1k_2} \mapsto U_{k_1l_1} U_{k_2l_2} (\mathbf{G} \otimes \mathbf{G})_{ij;;l_1l_2}$.

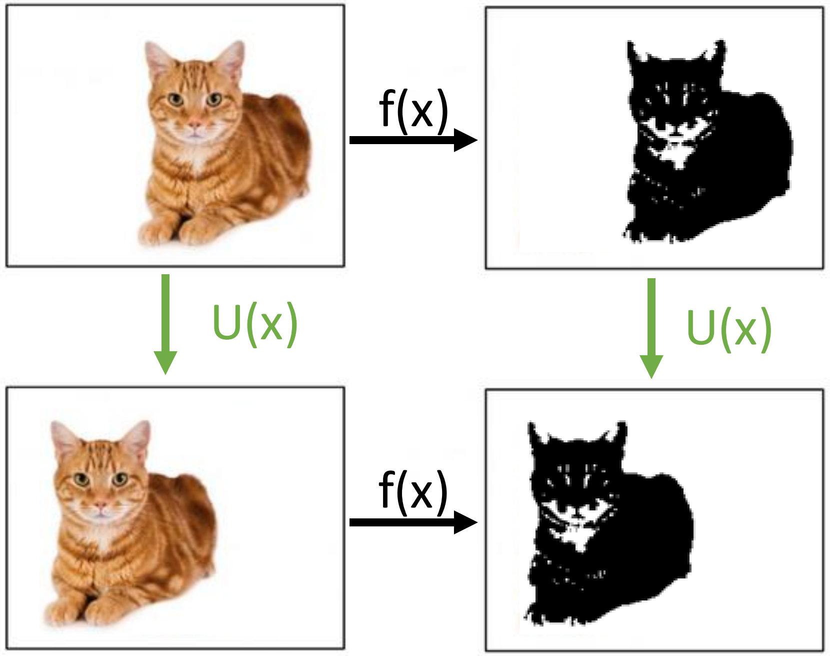
- $T{:}\ G \otimes G \longmapsto \sum U_{kn} G_{ij;;n} U_{mo} G_{jl;;o}$

$$= \sum U_{kn} G_{ij;;n} G_{jl;;o} U^T{}_{om}$$

Coordinate transformation

Proposition 3

$$(\mathbf{G} \otimes \mathbf{G})_{ij;;k_1k_2} \mapsto \boxed{U_{k_1l_1}U_{k_2l_2}}(\mathbf{G} \otimes \mathbf{G})_{ij;;l_1l_2}$$



Generalizing Prop. 3

$$\bigotimes^0 \mathbf{G} = 1$$
$$\bigotimes^1 \mathbf{G} = \mathbf{G}$$
$$\bigotimes^p \mathbf{G} = \bigotimes^{p-1} \mathbf{G} \otimes \mathbf{G}$$

$$\left[\left(\bigotimes^p \mathbf{G}\right) * \mathbf{H}^{(0)}\right]_{i;g;k_1 k_2 \ldots k_p} = \sum_j \left(\bigotimes^p \mathbf{G}\right)_{ij;;k_1 k_2 \ldots k_p} H^{(0)}_{j;g;}$$

- Convolution of p^{th} tensor power of G and rank-0 H

IsoGCN **Invariant** Layer

$$\boldsymbol{H}_{\text{out}}^{(0)} = \sigma\left((\mathbf{G} \odot \mathbf{G})\, \boldsymbol{H}_{\text{in}}^{(0)} \boldsymbol{W}\right) \qquad \boldsymbol{L} := \mathbf{G} \odot \mathbf{G} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$$

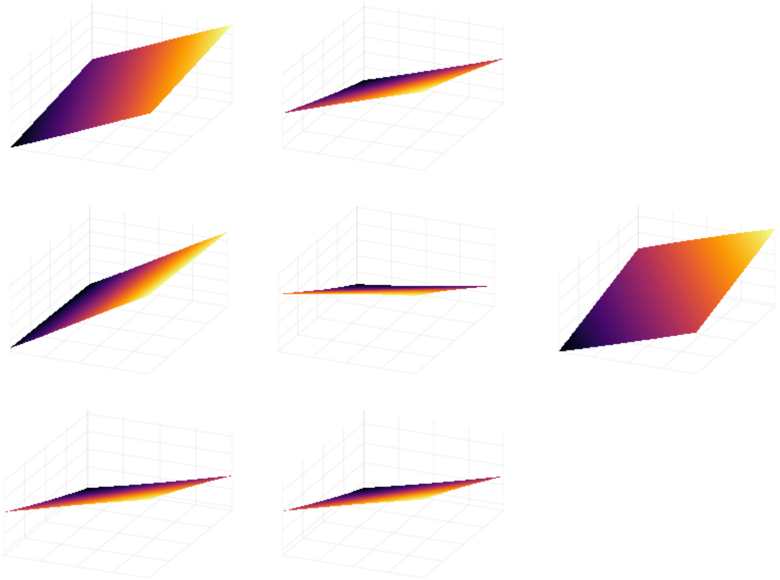
$$\boldsymbol{H}_{\text{out}}^{(0)} = \sigma\left(\boldsymbol{L} \boldsymbol{H}_{\text{in}}^{(0)} \boldsymbol{W}\right)$$

$$\boldsymbol{H}_{\text{out}} = \sigma(\hat{\boldsymbol{A}} \boldsymbol{H}_{\text{in}} \boldsymbol{W})$$

- W: trainable weights f_in x f_out

IsoGCN **Equivariant** Layer

- For tensor rank > 0: linear transformation → conv → tensor product
  - Non-linear activation / bias **distorts isometry**
  - But we must use non-linear activation, otherwise predictive performance is limited



IsoGCN **Equivariant** Layer

- For tensor rank > 0: linear transformation → conv → tensor product
  - Non-linear activation / bias distorts isometry
  - But we must use non-linear activation, otherwise predictive performance is limited

- **Convert input to rank-0 tensor** → apply non-linear activation → multiply with higher rank tensor

IsoGCN **Equivariant** Layer

Input tensor to rank-0
+ apply non-linearity

IsoTr equivariant
rank-m to rank-l learning

$$\mathbf{H}^{(l)}_{\mathrm{out}} = \boxed{F_{m\to 0}\left(\mathbf{H}^{(m)}_{\mathrm{in}}\right)} \times \boxed{\mathbf{F}_{m\to l}\left(\mathbf{H}^{(m)}_{\mathrm{in}}\right)}$$

Tensor product
→ IsoTr equivariant

$$\mathbf{F}_{m\to l}\left(\mathbf{H}^{(m)}_{\mathrm{in}}\right) = \boxed{\left[\bigotimes^{m-l} \mathbf{G}\right]} \odot \boxed{\mathbf{H}^{(m)}_{\mathrm{in}} W^{ml}}$$

$$\mathbf{H}^{(l)}_{\mathrm{out}} = \mathbf{H}^{(l)}_{\mathrm{in}} W + \sum_{m=0}^{M} f_{\mathrm{gather}}\left(\boxed{\left\{F_{k\to 0}(\mathbf{H}^{(k)}_{\mathrm{in}})\right\}_{k=0}^{M}}\right) \times \boxed{\mathbf{F}_{m\to l}\left(\mathbf{H}^{(m)}_{\mathrm{in}}\right)}$$

IsoGCN vs TFN

$$\mathbf{H}_{\text{out}}^{(l)} = \mathbf{H}_{\text{in}}^{(l)} W + \sum_{m=0}^{M} f_{\text{gather}} \left(\left\{ F_{k \to 0}(\mathbf{H}_{\text{in}}^{(k)}) \right\}_{k=0}^{M} \right) \times \left[\bigotimes^{m-l} \mathbf{G} \right] \odot \mathbf{H}_{\text{in}}^{(m)} W^{ml}$$

Tensor Field Networks:

$$\tilde{\mathbf{H}}_{\text{out},i}^{(l)} = w^{ll}\, \tilde{\mathbf{H}}_{\text{in},i}^{(l)} + \sum_{k \geq 0} \sum_{j \neq i} W^{lk} (\boldsymbol{x}_j - \boldsymbol{x}_i)\, \tilde{\mathbf{H}}_{\text{in},j}^{(k)}$$

IsoGCN **Equivariant** Layer

$$\mathbf{H}_{\text{out}}^{(l)} = \mathbf{H}_{\text{in}}^{(l)} W + \sum_{m=0}^{M} f_{\text{gather}} \left(\left\{ F_{k \to 0}(\mathbf{H}_{\text{in}}^{(k)}) \right\}_{k=0}^{M} \right) \times \mathbf{F}_{m \to l} \left(\mathbf{H}_{\text{in}}^{(m)} \right)$$

- Above layer as-is is only translation invariant
- Define reference vertex → compute output → add reference vertex

IsoAM for Mesh Structures

$$\tilde{D}_{ij;;k} = D_{ij;;k} - \delta_{ij} \sum_{l} D_{il;;k}$$

$$D_{ij;;:} = M_i^{-1} \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2} w_{ij} A_{ij}(m)$$

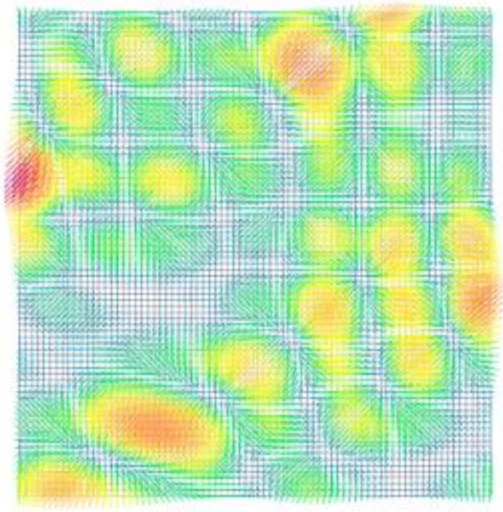
$$\mathbb{R}^d \ni \mathbf{G}_{ij;;:} := \sum_{k,l \in \mathcal{V}, k \neq l} T_{ijkl} (\boldsymbol{x}_k - \boldsymbol{x}_l)$$

$$M_i = \sum_{l} \frac{\boldsymbol{x}_l - \boldsymbol{x}_i}{\|\boldsymbol{x}_l - \boldsymbol{x}_i\|} \otimes \frac{\boldsymbol{x}_l - \boldsymbol{x}_i}{\|\boldsymbol{x}_l - \boldsymbol{x}_i\|} w_{il} A_{il}(m)$$

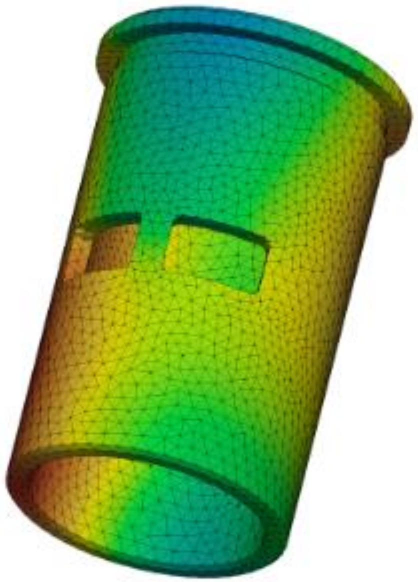
- D: IsoAM instance |V| x |V| x d
- A(m): Sparse adjacency matrix of *m*-hops
- w: Task-dependent untrainable weight

2 Experiments

1. Differential Operator



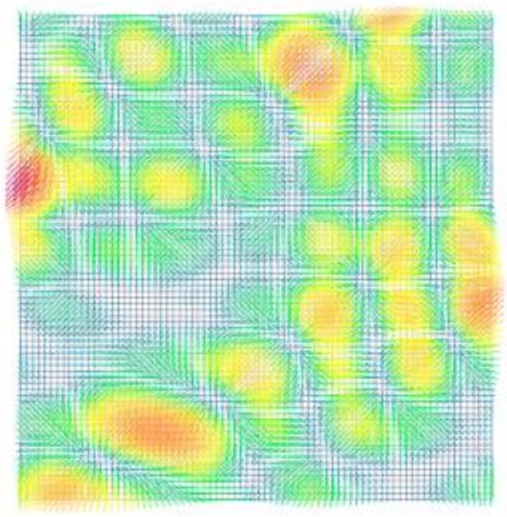
2. Anisotropic Heat Equation



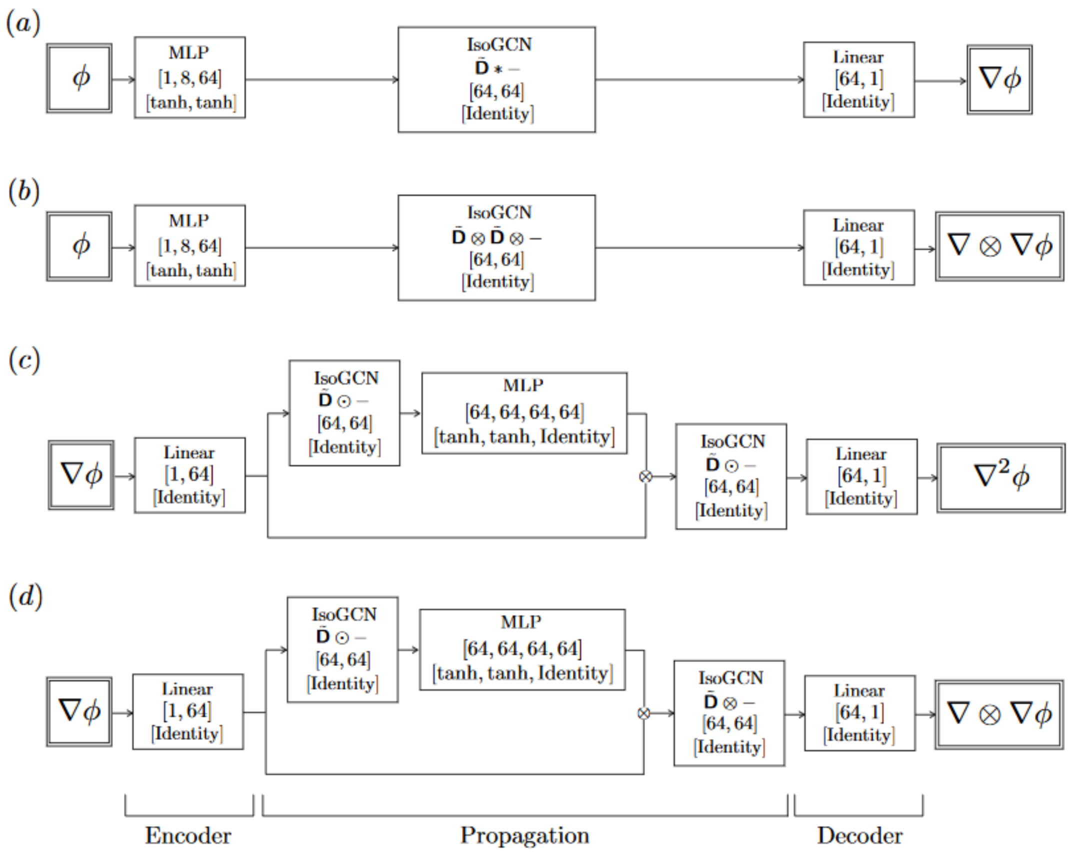
Differential Operator Dataset

- Generated pseudo-2D grid mesh
  - 1 cell in Z direction
  - 10-100 cells in X-Y directions
  - Generated scalar fields on the grid meshes
  - Calculated gradient, Laplacian, Hessian fields

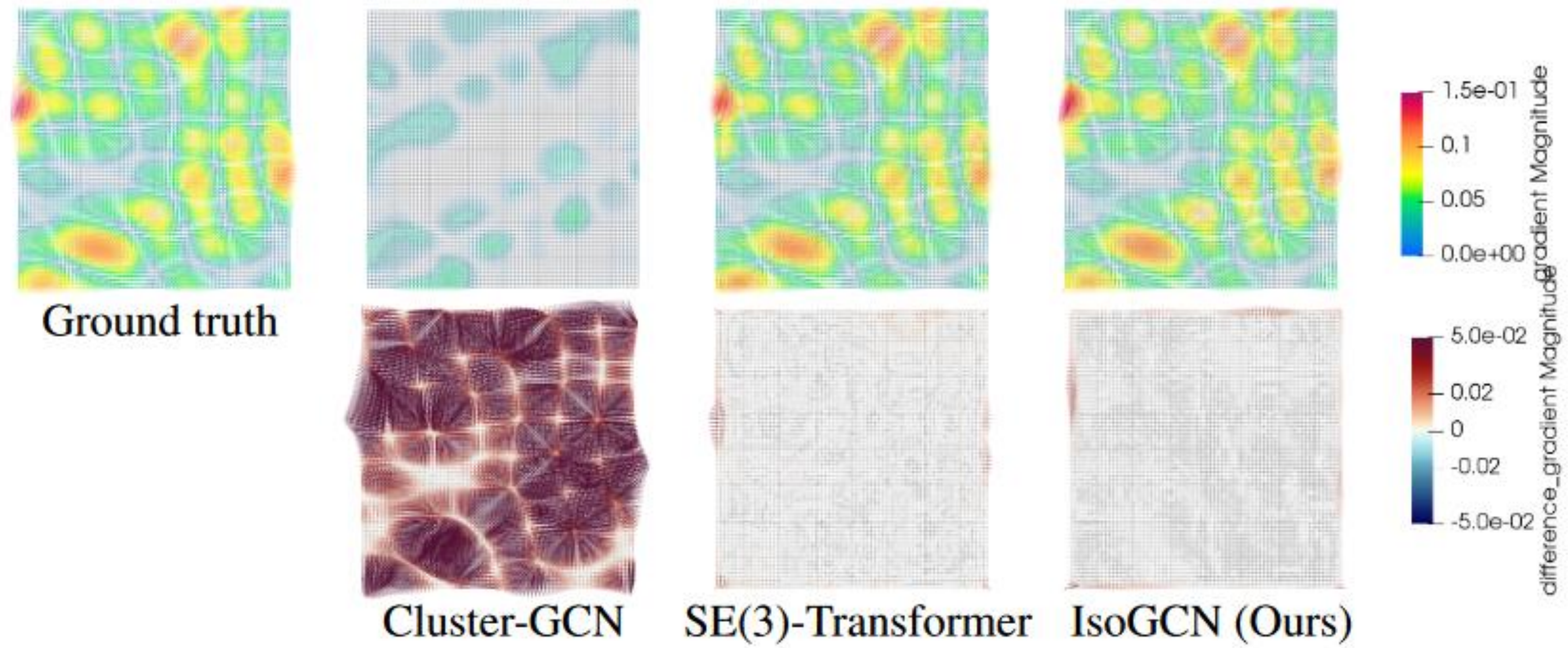
- 4 tasks
  - Scalar → gradient
  - Scalar → Hessian
  - Gradient → Laplacian
  - Gradient → Hessian



Differential Operator Dataset



Differential Operator (Results)



Differential Operator (Results)

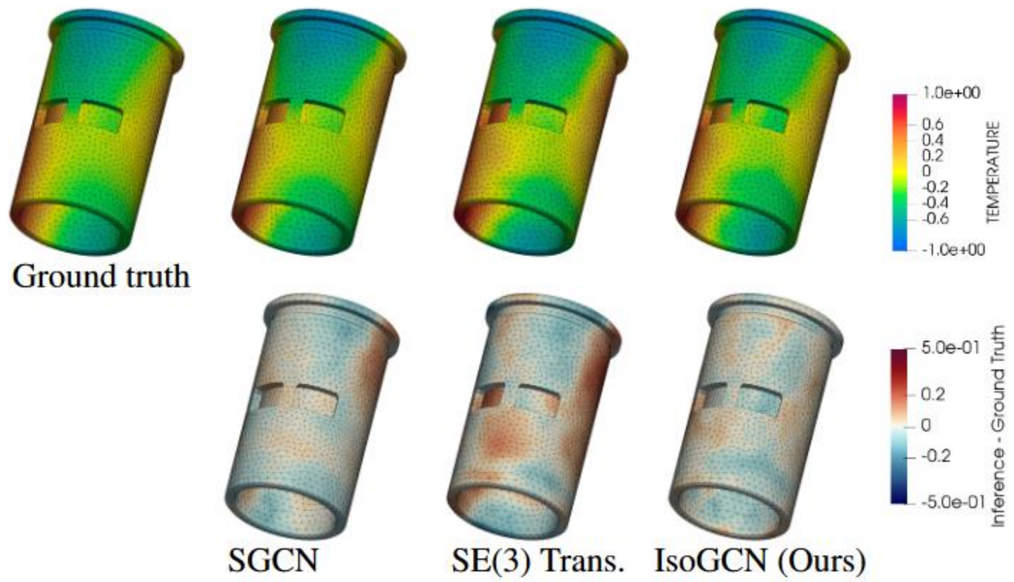
Method	# hops	x	Loss of $0 \to 1$ $\times 10^{-5}$	Loss of $0 \to 2$ $\times 10^{-6}$	Loss of $1 \to 0$ $\times 10^{-6}$	Loss of $1 \to 2$ $\times 10^{-6}$
GIN	5	Yes	147.07 ± 0.51	47.35 ± 0.35	404.92 ± 1.74	46.18 ± 0.39
GCNII	5	Yes	151.13 ± 0.53	31.87 ± 0.22	280.61 ± 1.30	39.38 ± 0.34
SGCN	5	Yes	151.16 ± 0.53	55.08 ± 0.42	127.21 ± 0.63	56.97 ± 0.44
GCN	5	Yes	151.14 ± 0.53	48.50 ± 0.35	542.30 ± 2.14	25.37 ± 0.28
Cluster-GCN	5	Yes	146.91 ± 0.51	26.60 ± 0.19	185.21 ± 0.99	18.18 ± 0.20
TFN	2	No	2.47 ± 0.02	OOM	26.69 ± 0.24	OOM
	5	No	OOM	OOM	OOM	OOM
SE(3)-Trans.	2	No	$\mathbf{1.79} \pm 0.02$	$\mathbf{3.50} \pm 0.04$	$\mathbf{2.52} \pm 0.02$	OOM
	5	No	2.12 ± 0.02	OOM	7.66 ± 0.05	OOM
IsoGCN (Ours)	2	No	2.67 ± 0.02	6.37 ± 0.07	7.18 ± 0.06	$\mathbf{1.44} \pm 0.02$
	5	No	14.19 ± 0.10	21.72 ± 0.25	34.09 ± 0.19	8.32 ± 0.09

Differential Operator (Results)

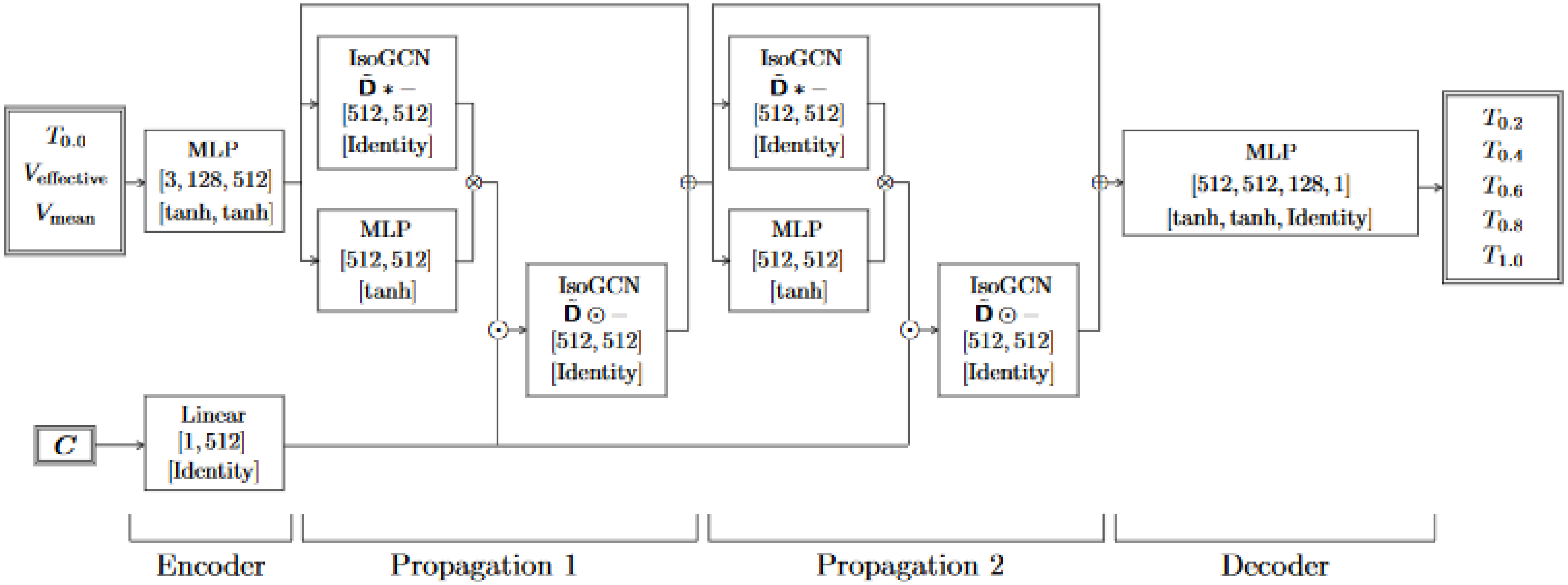
	$\mathbf{0 \to 1}$		$\mathbf{0 \to 2}$	
Method	**# parameters**	**Inference time [s]**	**# parameters**	**Inference time [s]**
TFN	5264	3.8	5220	OOM
SE(3)-Trans.	5392	4.0	5265	9.2
IsoGCN (Ours)	4816	0.4	4816	0.7

Anisotropic Nonlinear Heat Equation

- Used CAD objects from ABC dataset
  - Generated first-order tetrahedral meshes (Gmsh)
  - Set temperature + anisotropic thermal conductivity → finite element analysis



Anisotropic Nonlinear Heat Equation



Anisotropic Nonlinear Heat Equation (Results)

Method	# hops	x	Loss $\times 10^{-3}$
GIN	2	No	16.921 ± 0.040
GCN	2	No	10.427 ± 0.028
GCNII	5	No	8.377 ± 0.024
Gluster-GCN	2	No	7.266 ± 0.021
SGCN	5	No	6.426 ± 0.018
TFN	2	No	15.661 ± 0.019
	5	No	OOM
SE(3)-Trans.	2	No	14.164 ± 0.018
	5	No	OOM
IsoGCN (Ours)	2	No	4.674 ± 0.014
	5	No	$\mathbf{2.470} \pm 0.008$

Anisotropic Nonlinear Heat Equation (Results)

Method	$\lvert\mathcal{V}\rvert = 21,289$ Loss $\times 10^{-4}$	Time [s]	$\lvert\mathcal{V}\rvert = 155,019$ Loss $\times 10^{-4}$	Time [s]	$\lvert\mathcal{V}\rvert = 1,011,301$ Loss $\times 10^{-4}$	Time [s]
FrontISTR ($\Delta t = 1.0$)	10.9	16.7	6.1	181.7	2.9	1656.5
FrontISTR ($\Delta t = 0.5$)	0.8	30.5	0.4	288.0	0.2	2884.2
TFN	77.9	46.1	30.1	400.9	OOM	OOM
SE(3)-Transformer	111.4	31.2	80.3	271.1	OOM	OOM
IsoGCN (Ours)	8.1	**7.4**	4.9	**84.1**	3.9	**648.4**

Conclusion

- Main application of IsoGCN: **learning physical simulations**
  - Very short computation time
  - Isometric transformation invariant + equivariant
- More memory efficient than TFN and SE(3)-Transformer

Limitations

- Assumptions of IsoGCN
    1. Attributes are associated with vertices and not edges
    2. Graphs do not contain self-loops

- SE(3) Transformer did better in differential operator experiment
    - Doesn’t convincingly perform better than TFN either

- No experiments with isometric transformation
    - Only proven via propositions / proofs