物理シミュレーションのための
同変グラフニューラルネットワーク

Deep learning and Physics

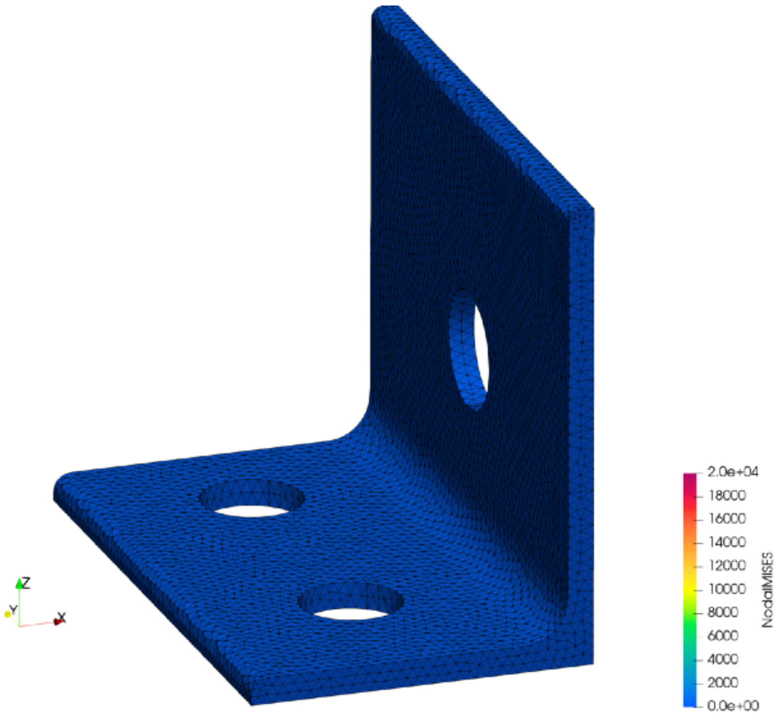
2021 年 4 月 22 日

堀江正信

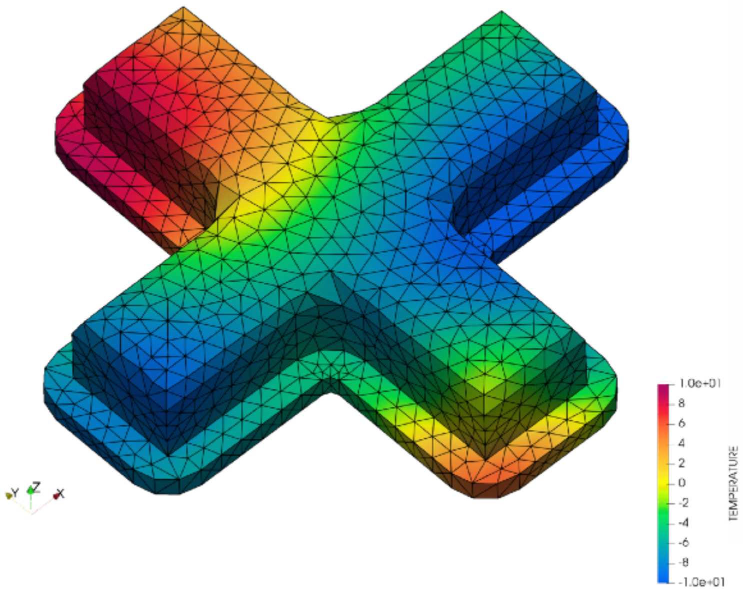
株式会社科学計算総合研究所

筑波大学システム情報工学研究群

背景：物理シミュレーションを軽量化したい



構造



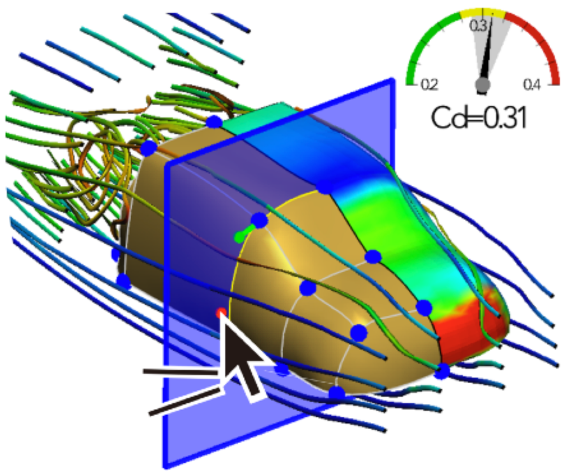
熱

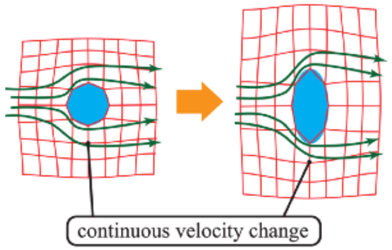
流体

- シミュレーションは製品の物理特性の評価において重要
- 計算時間が長く、シミュレーションを用いた最適化が困難な場合がある

➡ 軽量なモデル（サロゲートモデル）で重いシミュレーションを置き換えたい

関連研究：物理シミュレーションの機械学習

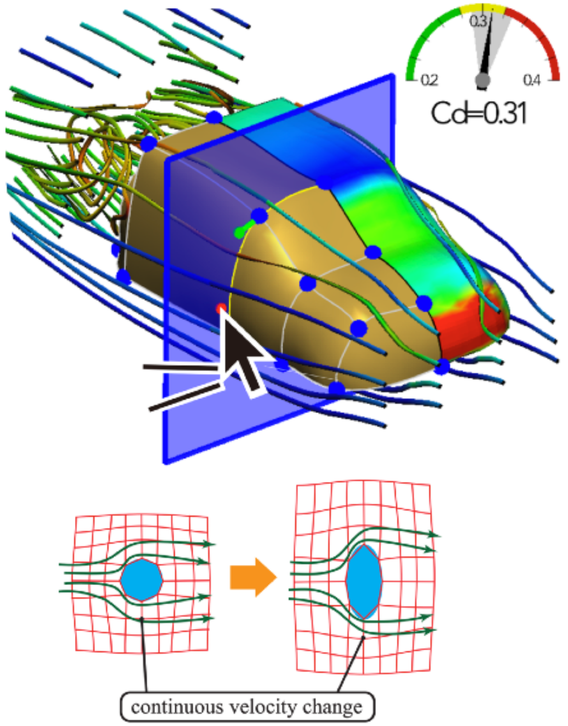




Umetani+ 2018

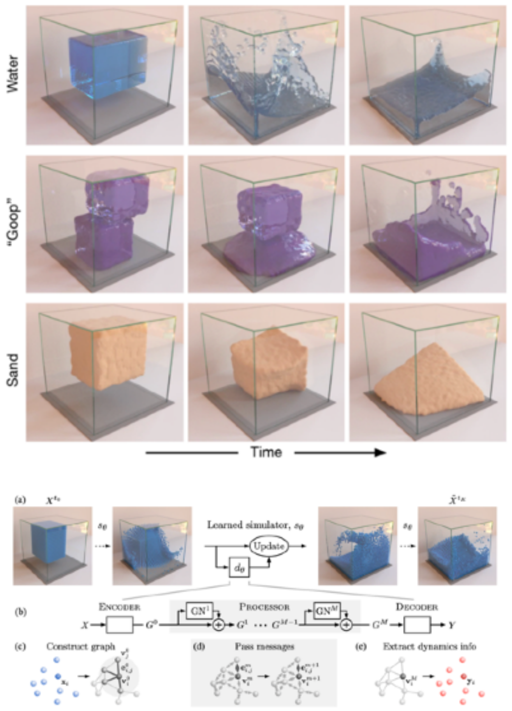
- リアルタイムで
  車体の空力性能予測
- メッシュ構造固定

関連研究：物理シミュレーションの機械学習



Umetani+ 2018

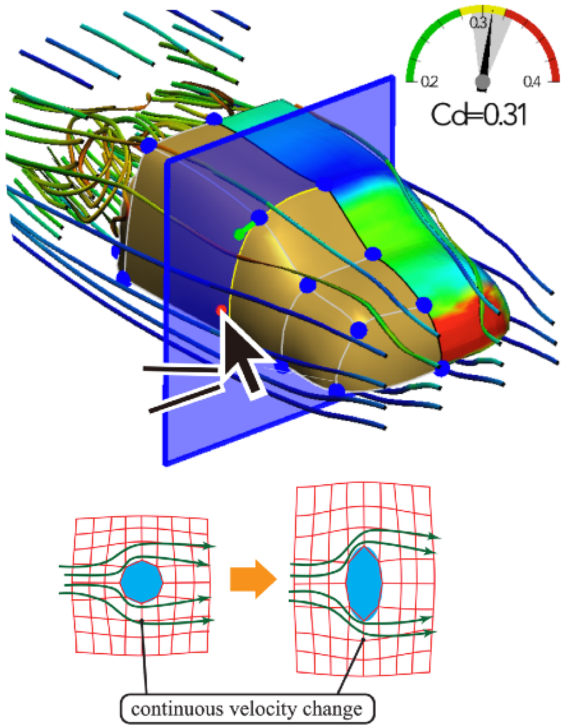
- リアルタイムで
  車体の空力性能予測
- メッシュ構造固定



Sanchez-Gonzalez+ 2020

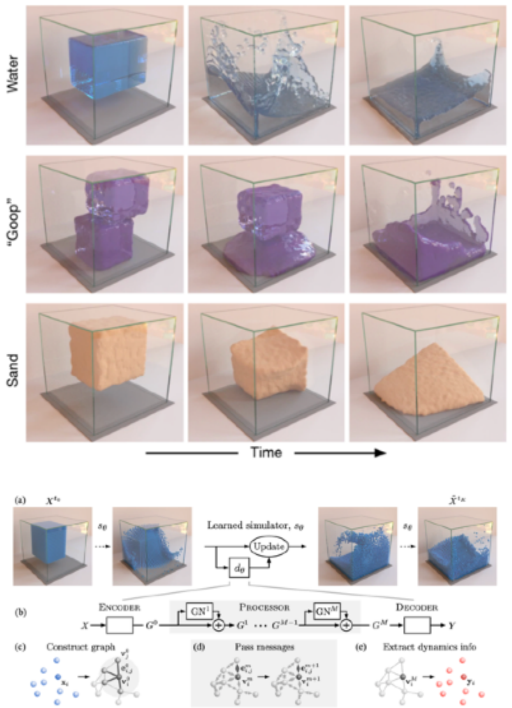
- 任意点群での学習
- 高速化は未達成

関連研究：物理シミュレーションの機械学習



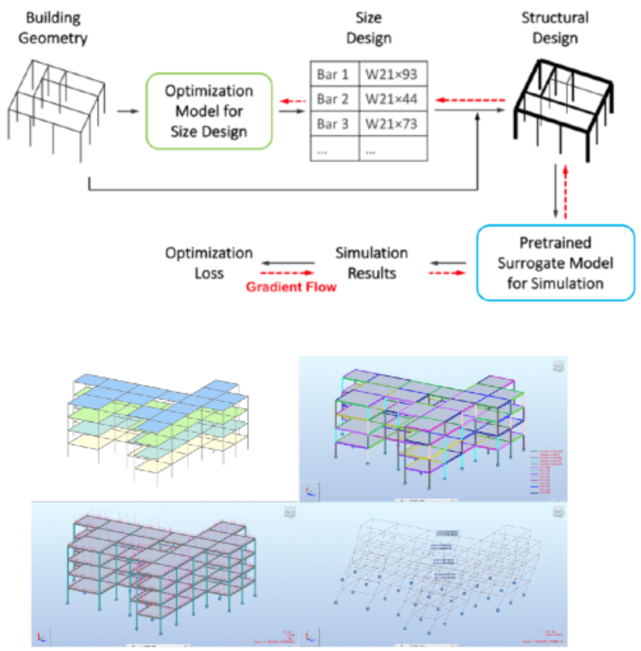
Umetani+ 2018

- リアルタイムで
  車体の空力性能予測
- メッシュ構造固定



Sanchez-Gonzalez+ 2020

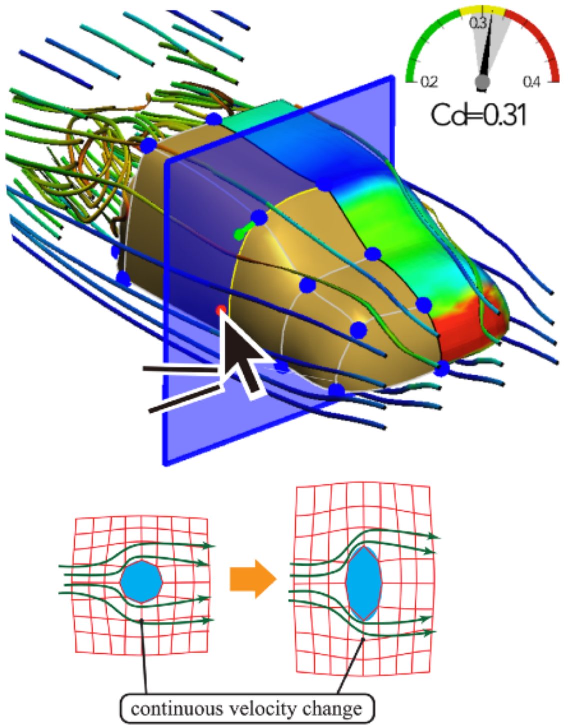
- 任意点群での学習
- 高速化は未達成



Chang+ 2020

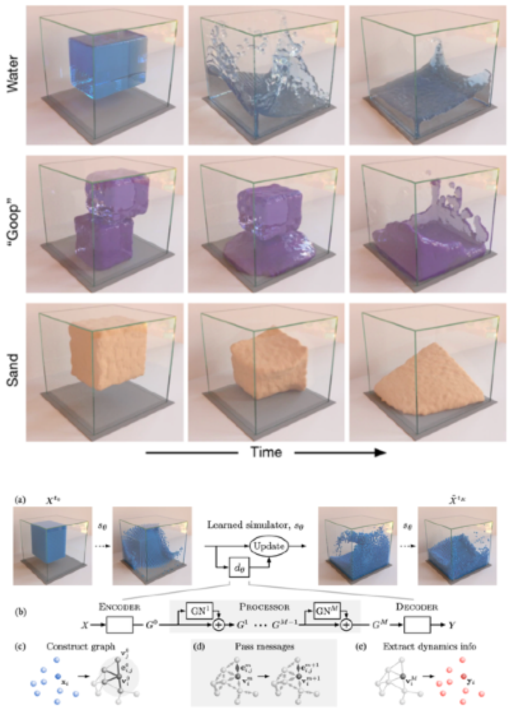
- 任意梁構造での学習
- 梁部材選定の最適化

関連研究：物理シミュレーションの機械学習



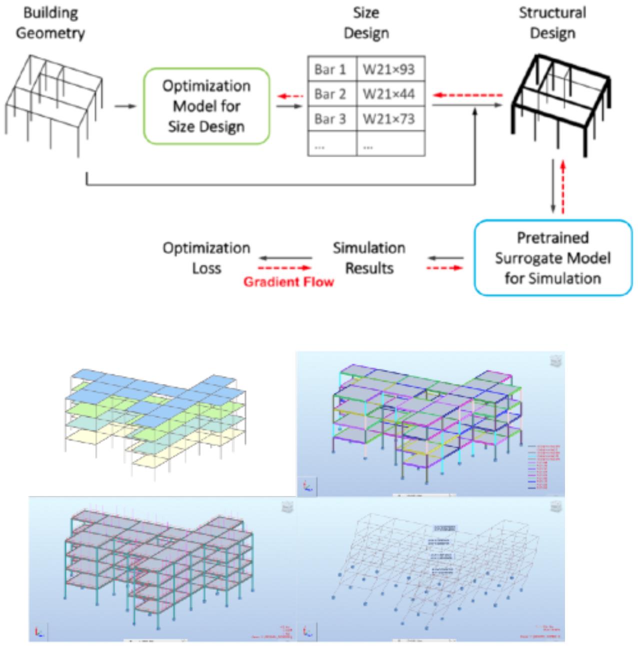
Umetani+ 2018

- リアルタイムで
  車体の空力性能予測
- メッシュ構造固定



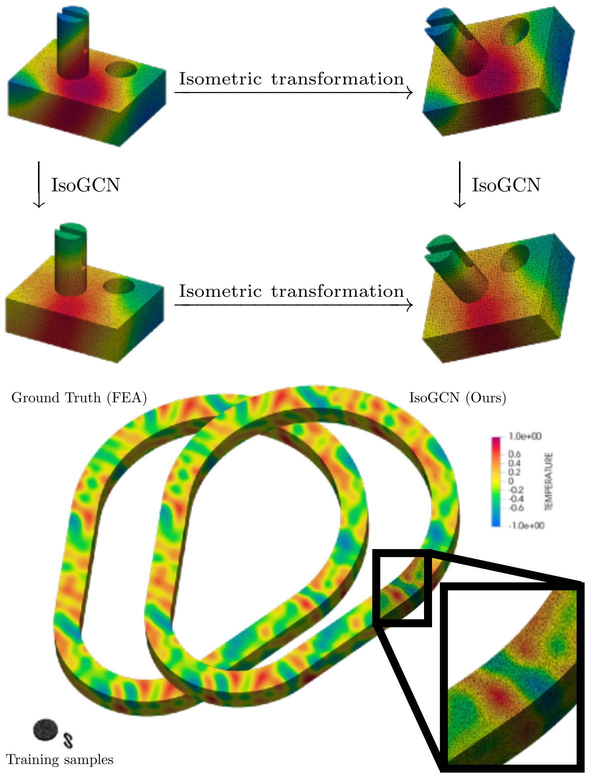
Sanchez-Gonzalez+ 2020

- 任意点群での学習
- 高速化は未達成



Chang+ 2020

- 任意梁構造での学習
- 梁部材選定の最適化

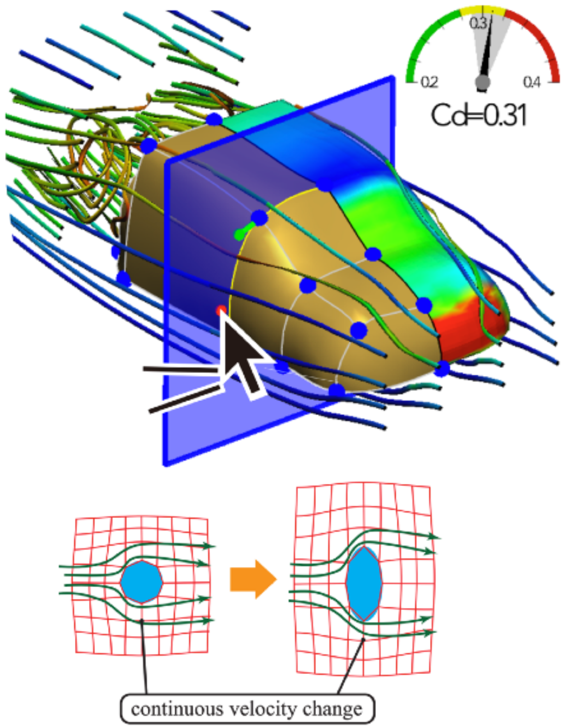


Horie+ 2020 (IsoGCN)

- 任意メッシュでの学習
- 高速化を達成
- 物理現象の対称性を考慮

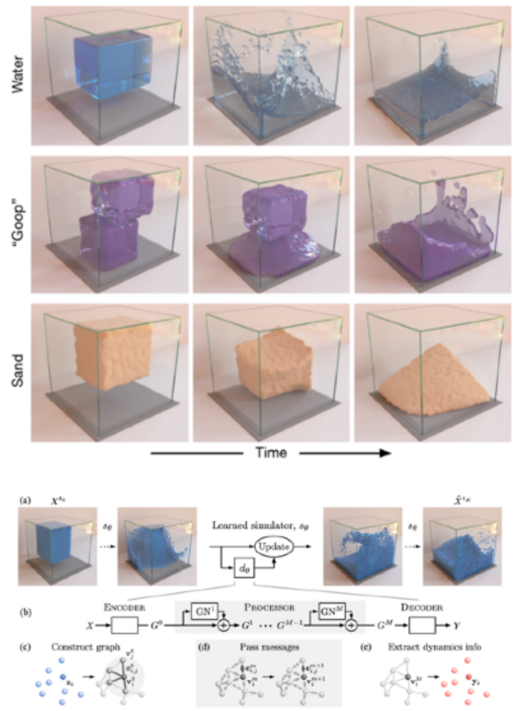
関連研究：物理シミュレーションの機械学習

→ Graph Neural Network の適用　　→ 同変性の導入



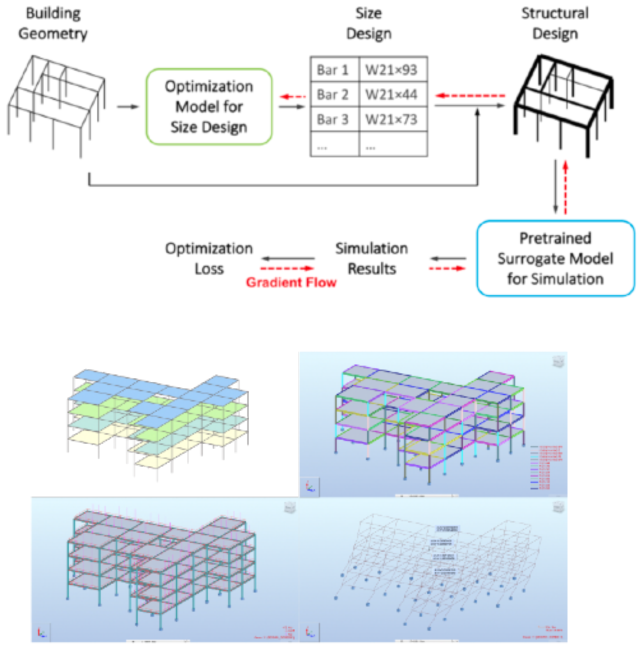
Umetani+ 2018

- リアルタイムで
  車体の空力性能予測
- メッシュ構造固定



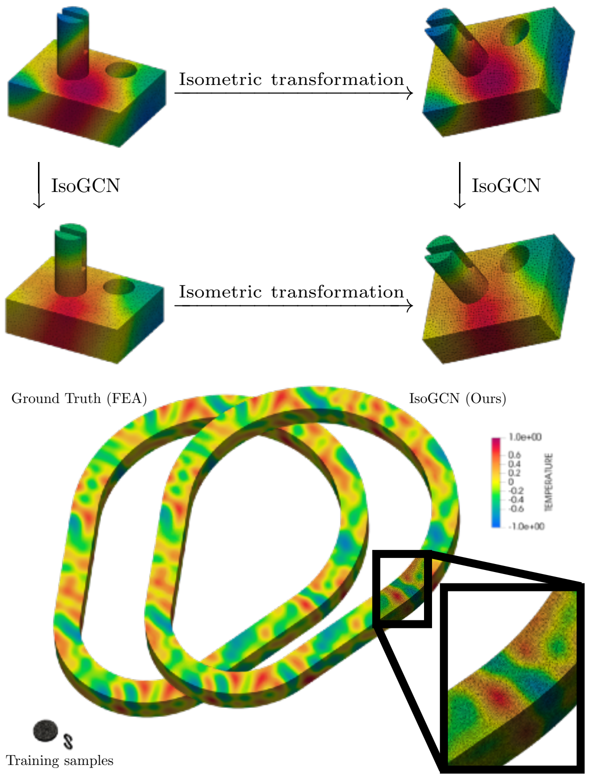
Sanchez-Gonzalez+ 2020

- 任意点群での学習
- 高速化は未達成



Chang+ 2020

- 任意梁構造での学習
- 梁部材選定の最適化



Horie+ 2020 (IsoGCN)

- 任意メッシュでの学習
- 高速化を達成
- 物理現象の対称性を考慮

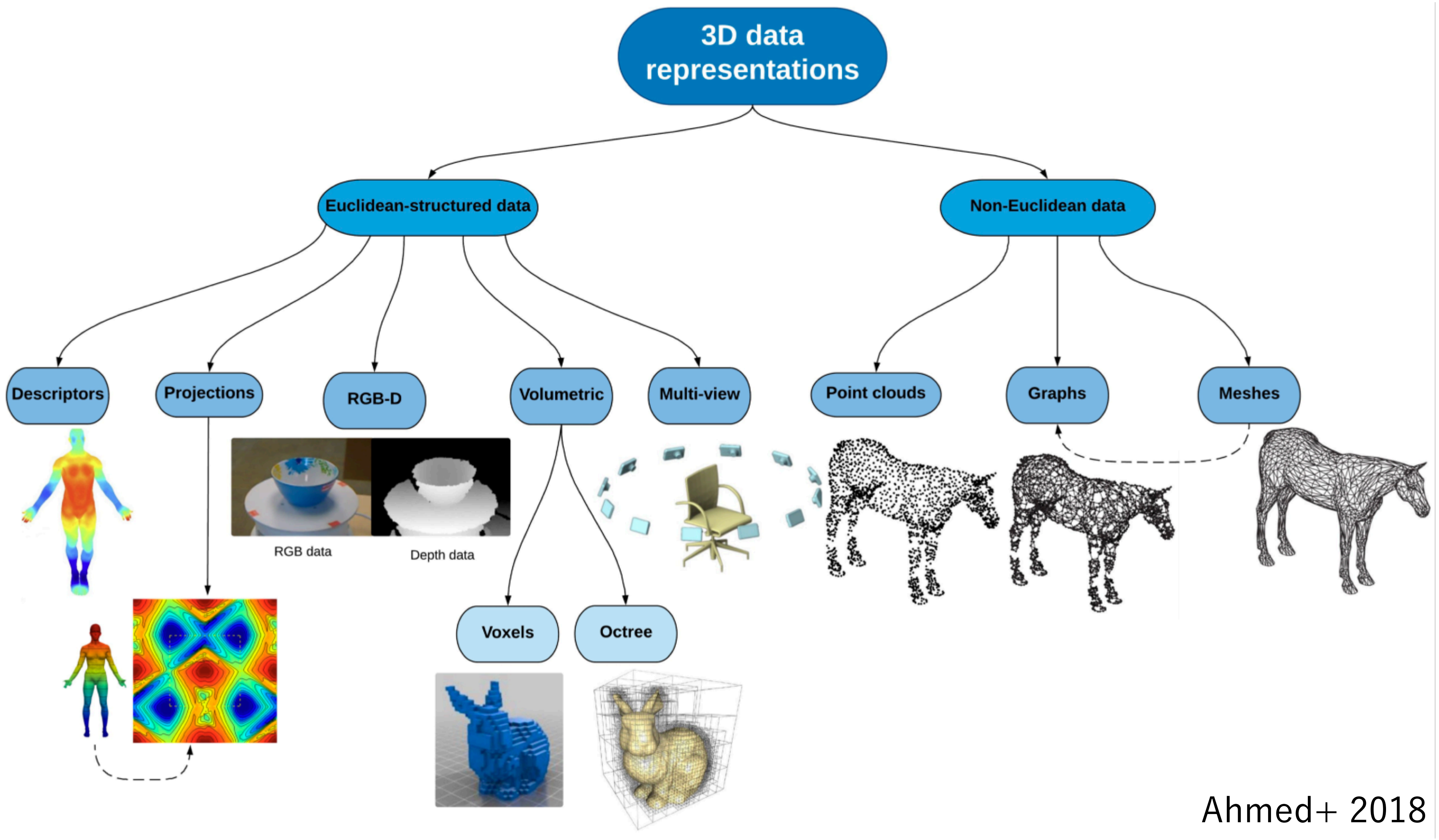
目次

- Graph neural network
- 同変性
- IsoGCN: 同変性＋Graph Neural Network
- まとめ

目次

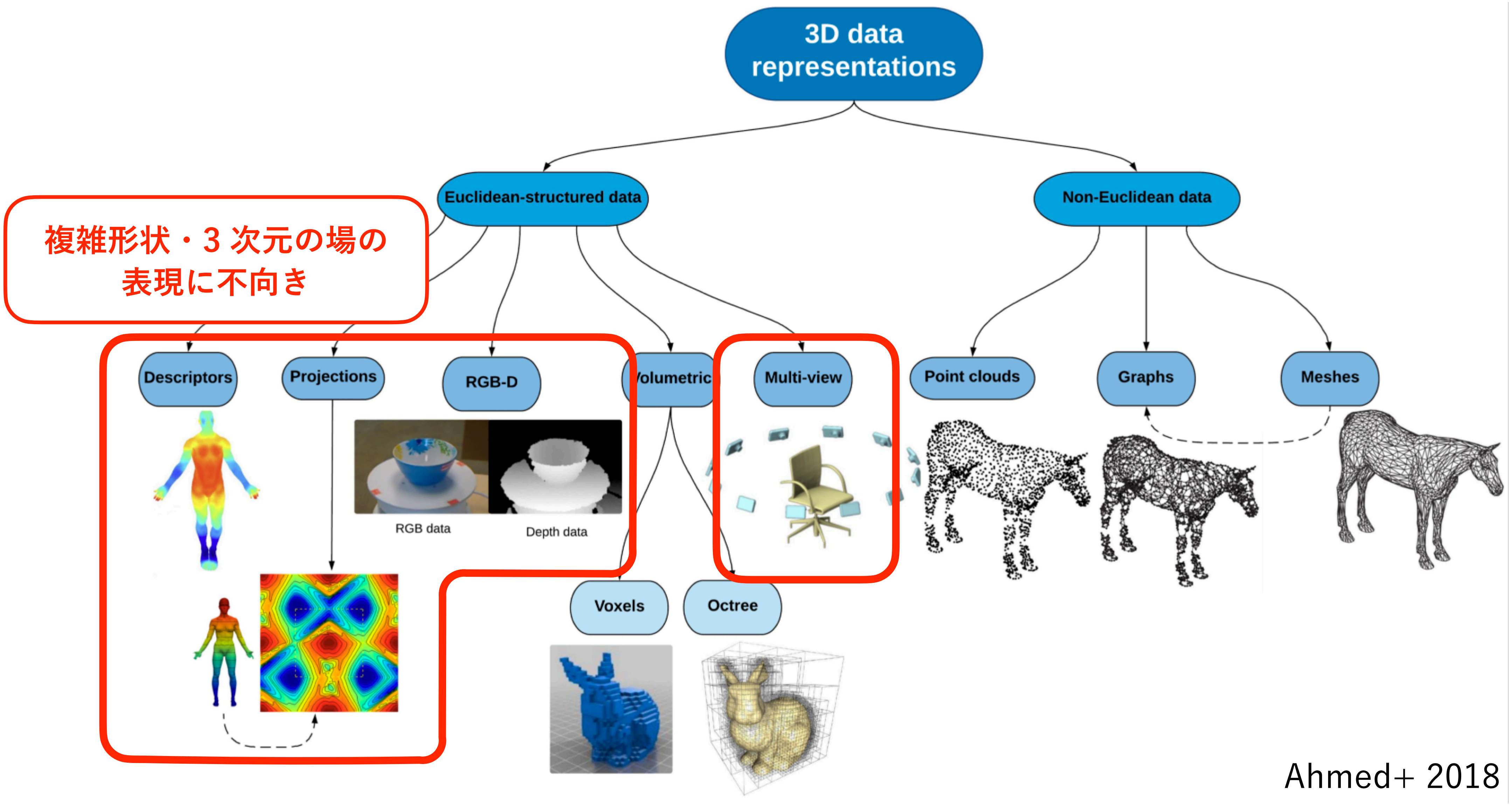
- Graph neural network
- 同変性
- IsoGCN: 同変性＋Graph Neural Network
- まとめ

3D データをどう扱うか？



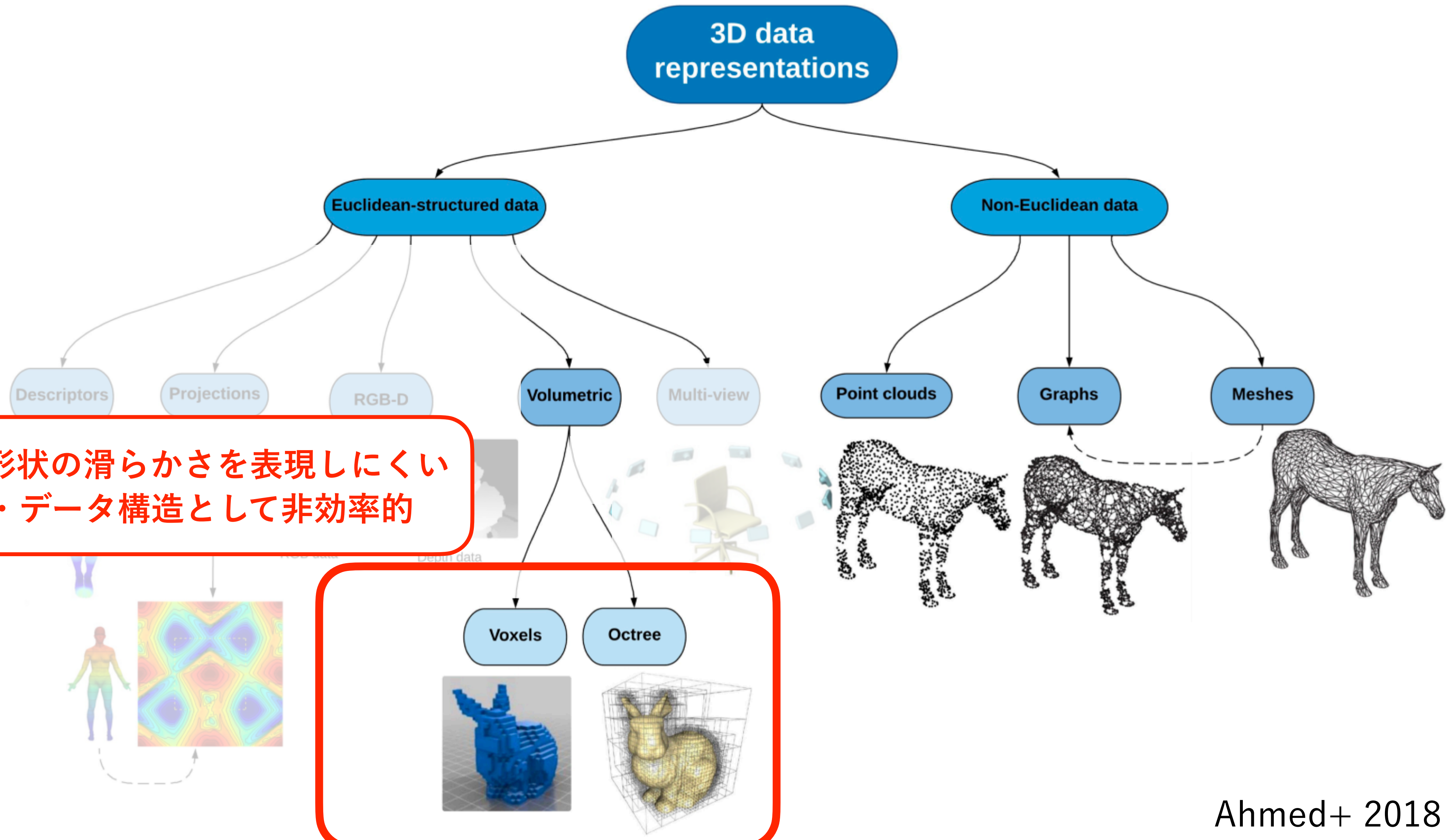
Ahmed+ 2018

3D データをどう扱うか？



Ahmed+ 2018

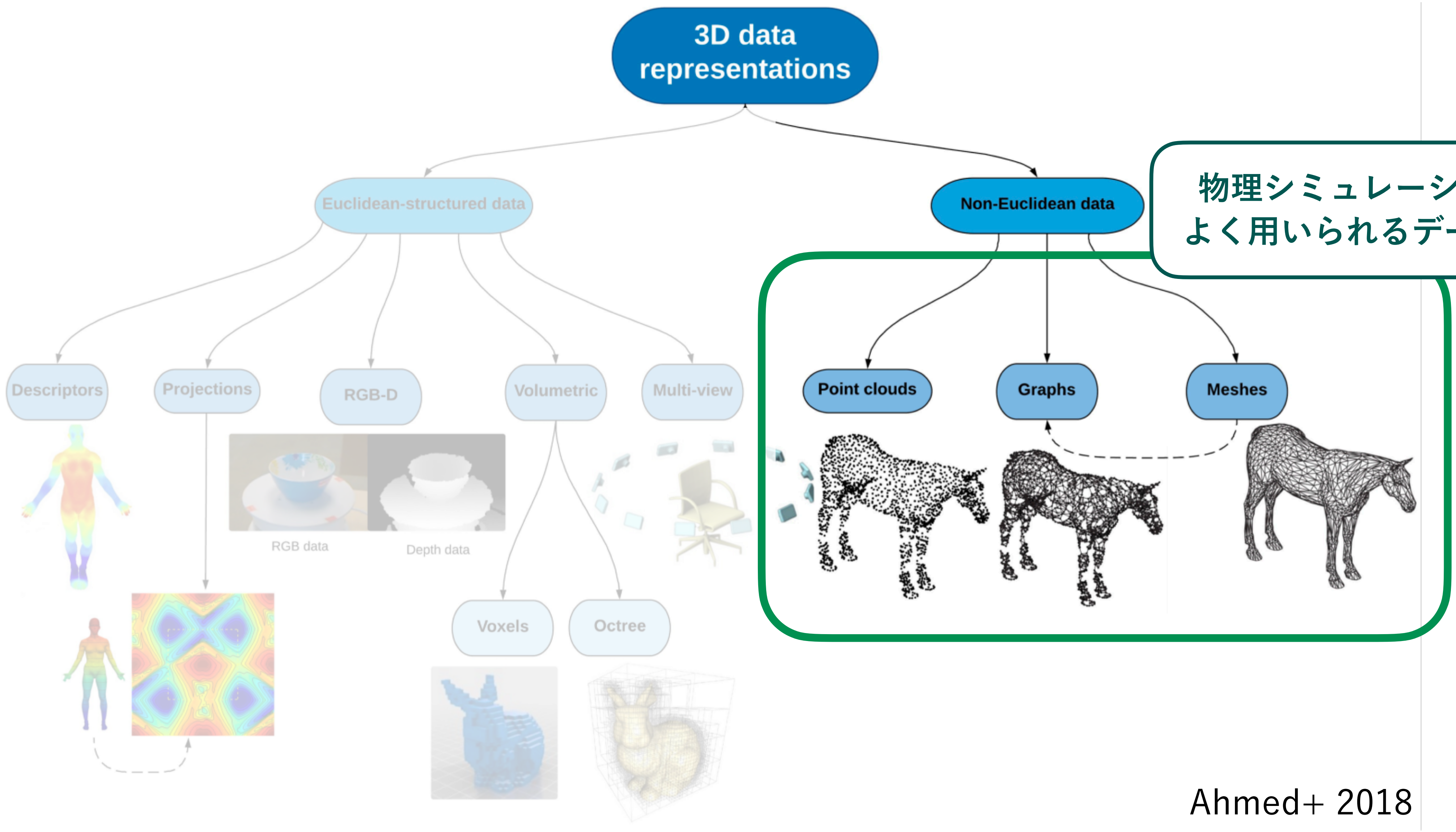
3D データをどう扱うか？



・形状の滑らかさを表現しにくい
・データ構造として非効率的

Ahmed+ 2018

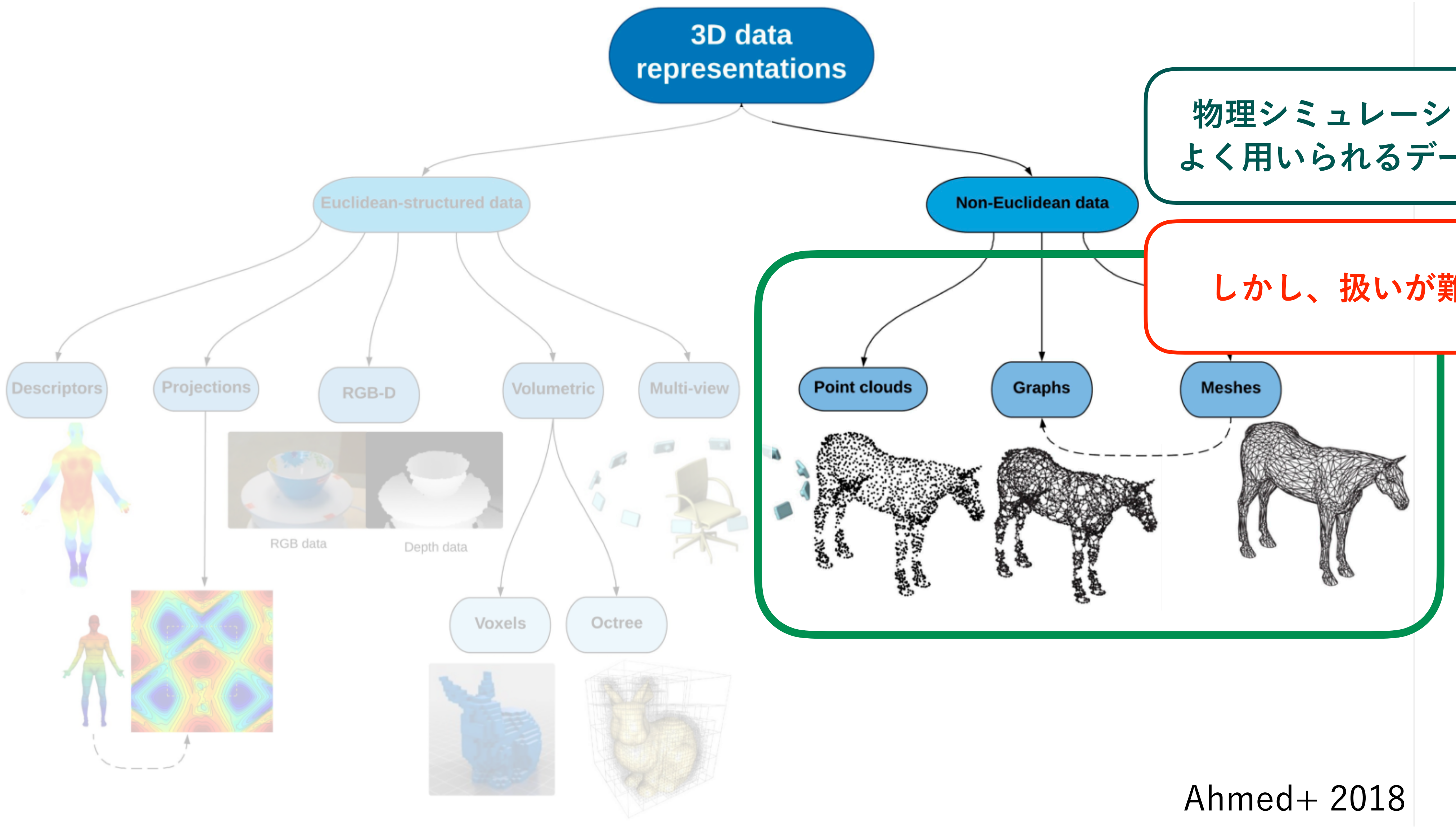
3D データをどう扱うか？



物理シミュレーションでよく用いられるデータ構造

Ahmed+ 2018

3D データをどう扱うか？



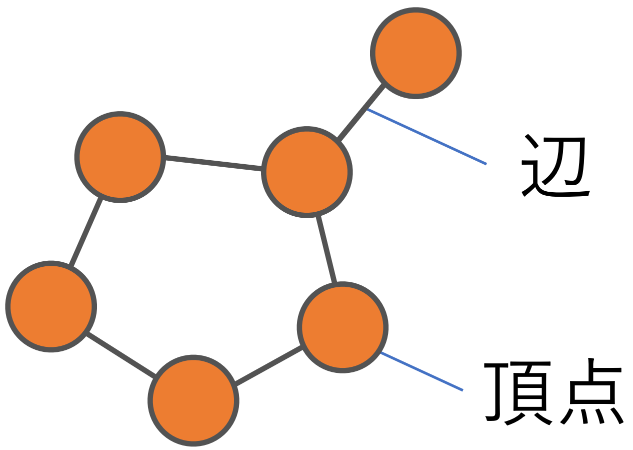
物理シミュレーションで
よく用いられるデータ構造

しかし、扱いが難しい

Ahmed+ 2018

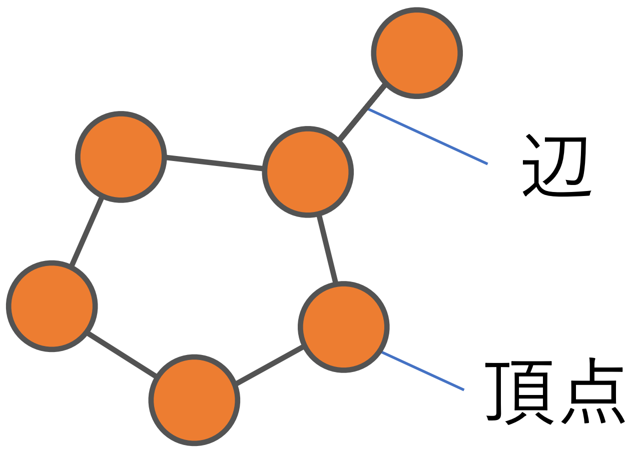
グラフの難しさ

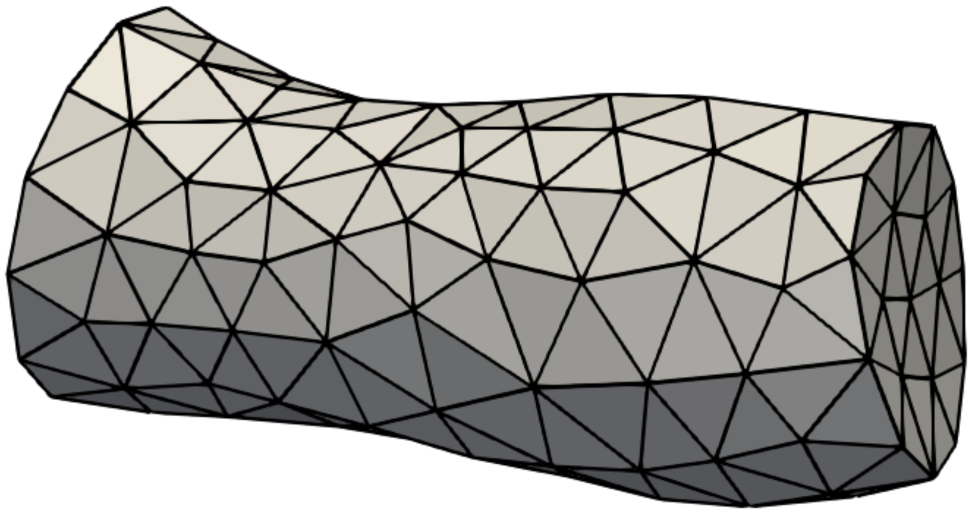
- グラフ：頂点（vertex）と辺（edge）からなるデータ構造



グラフの難しさ

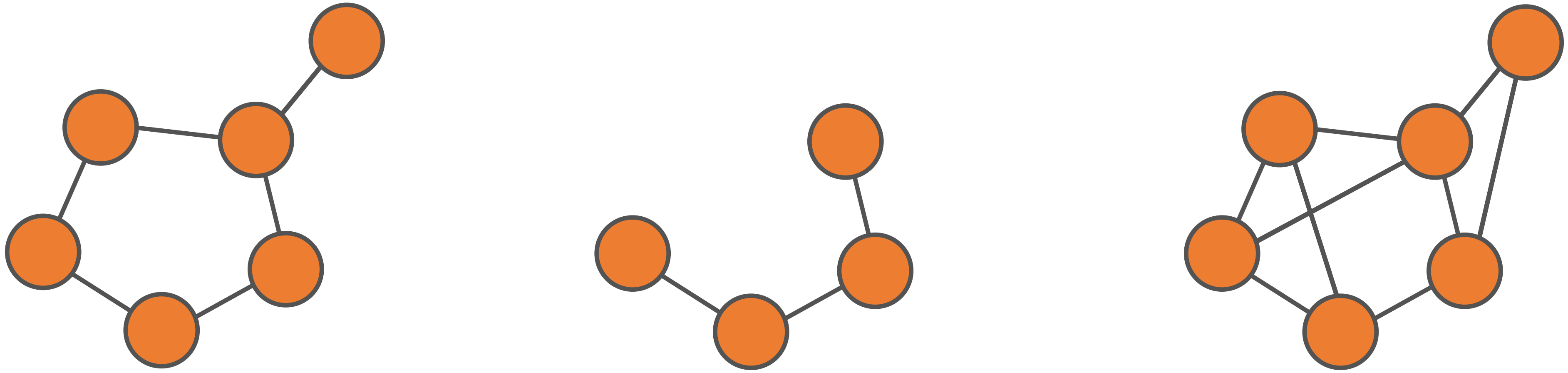
- グラフ：頂点（vertex）と辺（edge）からなるデータ構造
  - メッシュもグラフとみなせる





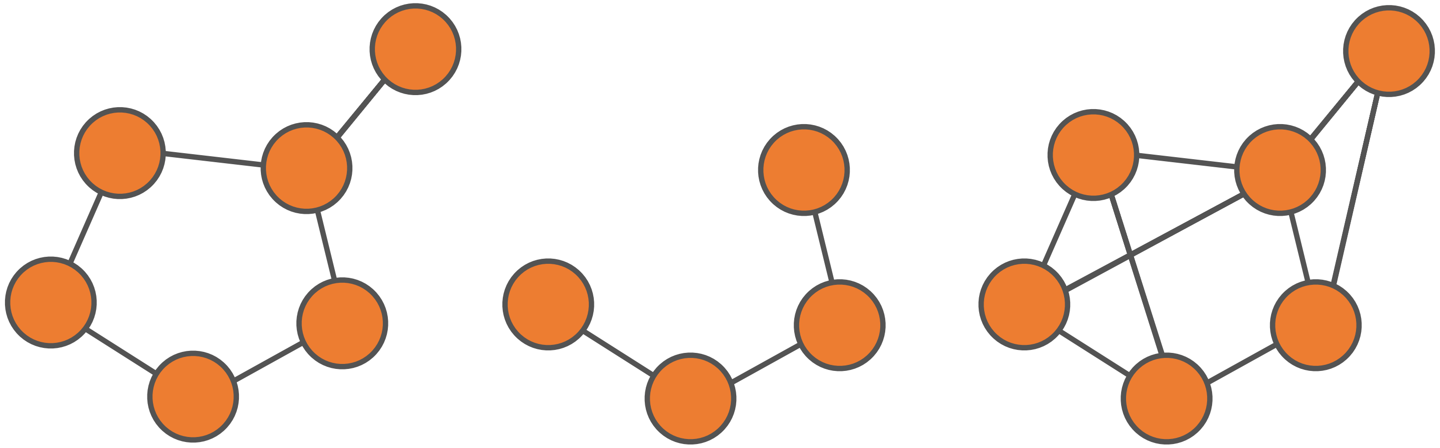
グラフの難しさ

- グラフ：頂点（vertex）と辺（edge）からなるデータ構造
  - メッシュもグラフとみなせる
- グラフの難しいところ：
  - 頂点の数が不定
  - 頂点から伸びているエッジの数が不定



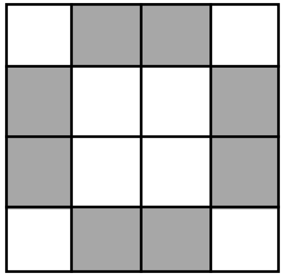
グラフの難しさ

- グラフ：頂点（vertex）と辺（edge）からなるデータ構造
  - メッシュもグラフとみなせる
- グラフの難しいところ：
  - 頂点の数が不定
  - 頂点から伸びているエッジの数が不定

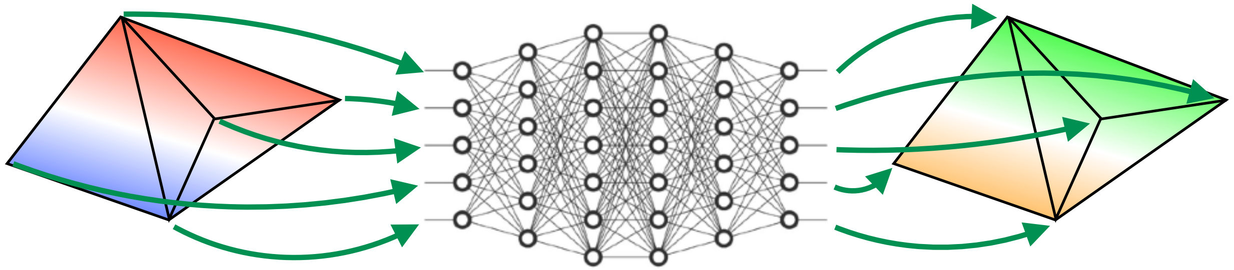


より扱いやすい例：

画像（格子グラフ）



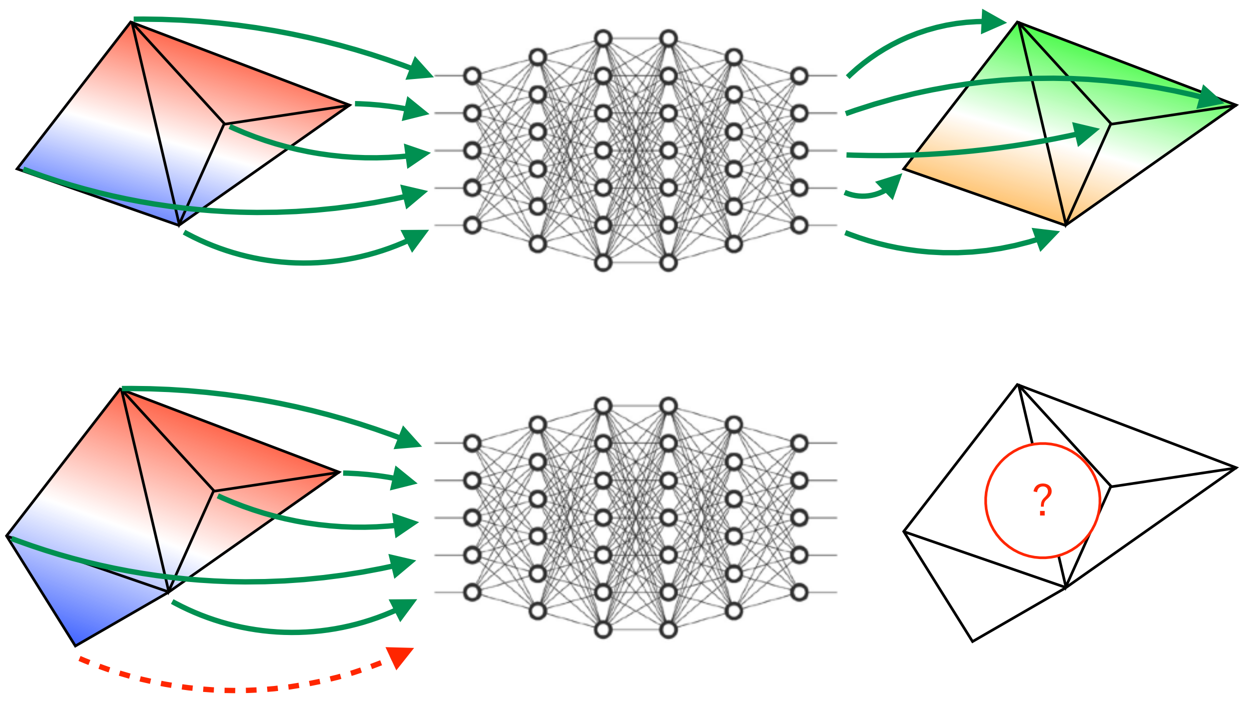
全結合ネットワークと Graph Neural Network



全結合ニューラルネットワーク

- 全結合のネットワークではグラフが変わると対応できない

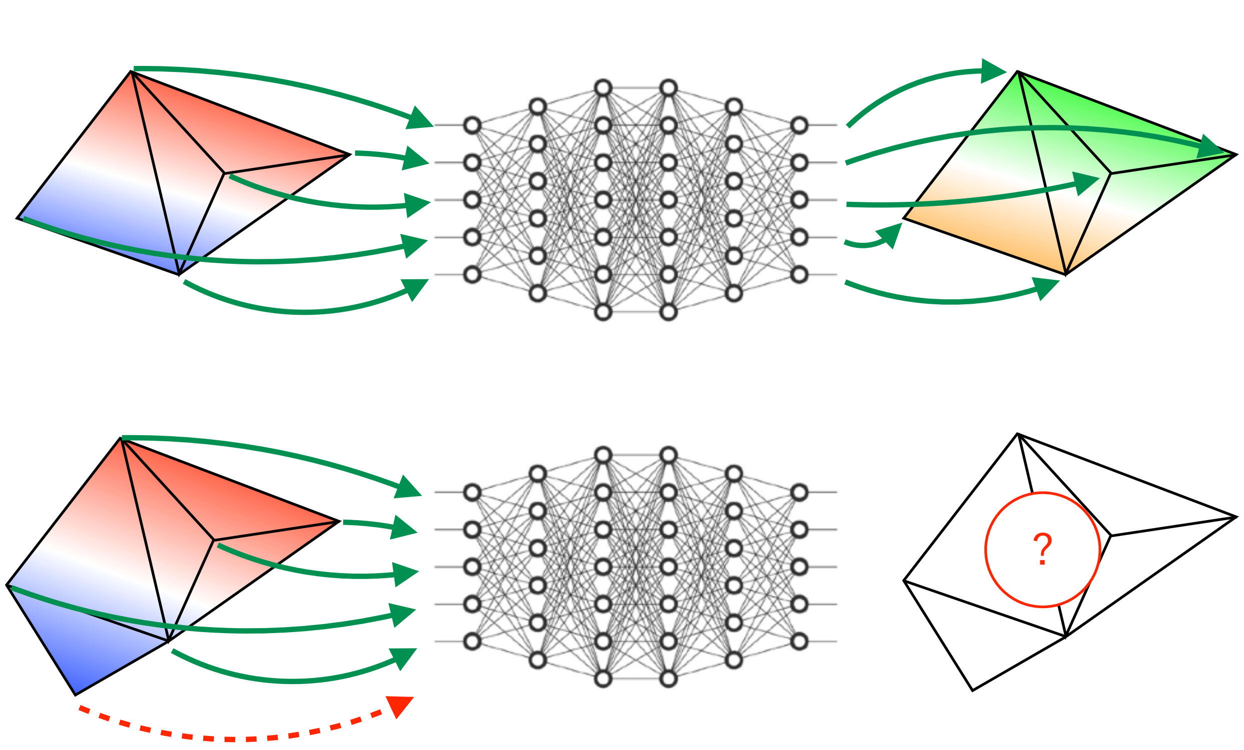
全結合ネットワークと Graph Neural Network



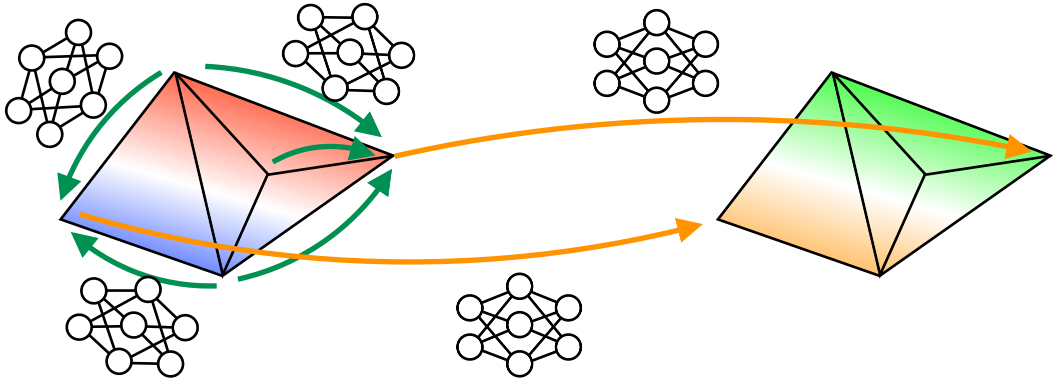
全結合ニューラルネットワーク

- 全結合のネットワークではグラフが変わると対応できない

全結合ネットワークと Graph Neural Network



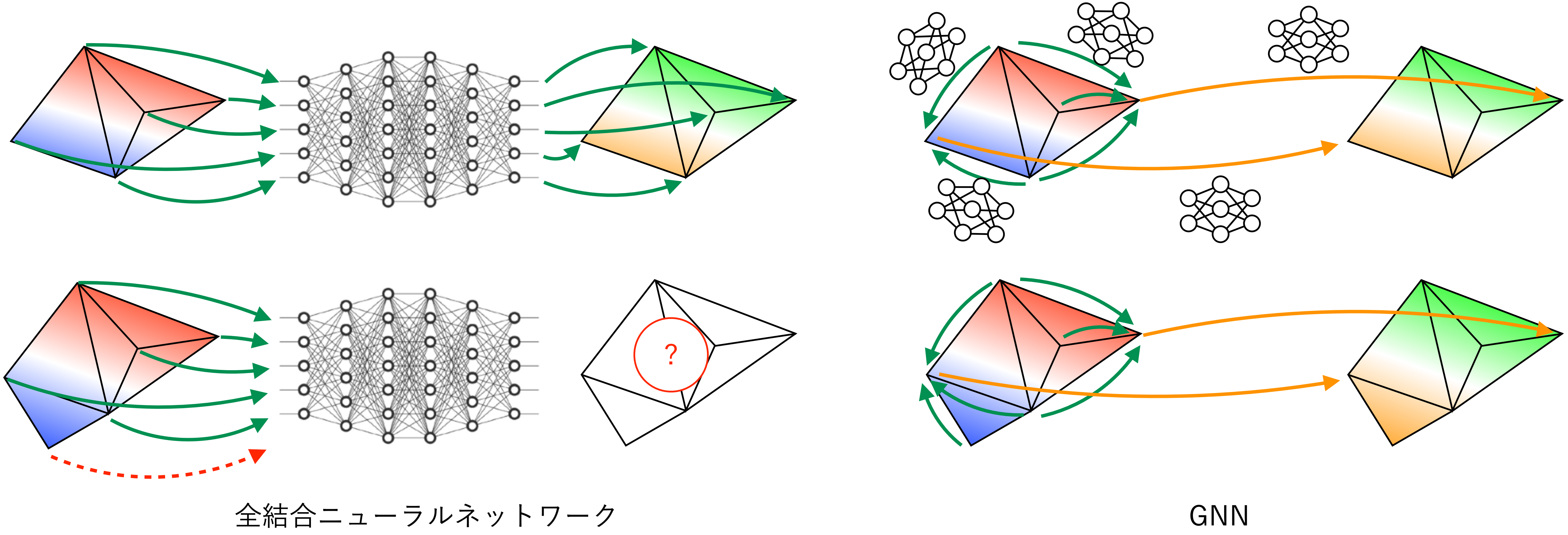
全結合ニューラルネットワーク



GNN

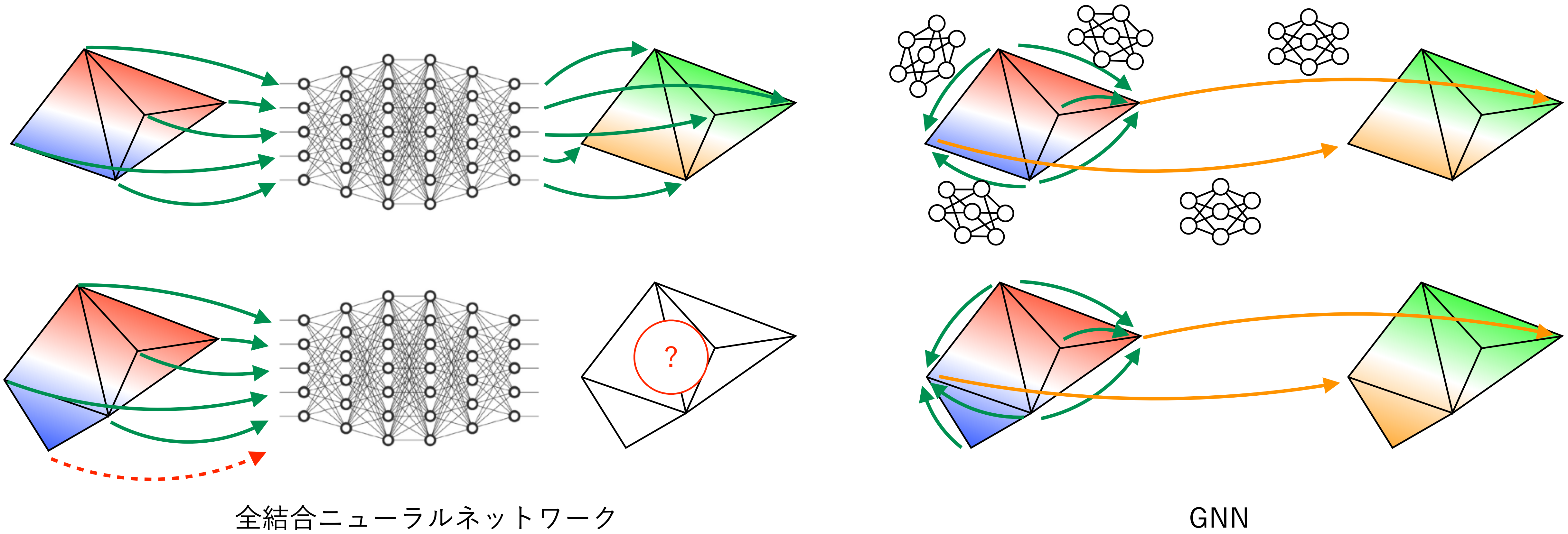
- 全結合のネットワークではグラフが変わると対応できない
- Graph Neural Network (GNN) はグラフの構造が変わっても同じモデルが使える

全結合ネットワークと Graph Neural Network



- 全結合のネットワークではグラフが変わると対応できない
- Graph Neural Network (GNN) はグラフの構造が変わっても同じモデルが使える

全結合ネットワークと Graph Neural Network



- 全結合のネットワークではグラフが変わると対応できない
- Graph Neural Network (GNN) はグラフの構造が変わっても同じモデルが使える

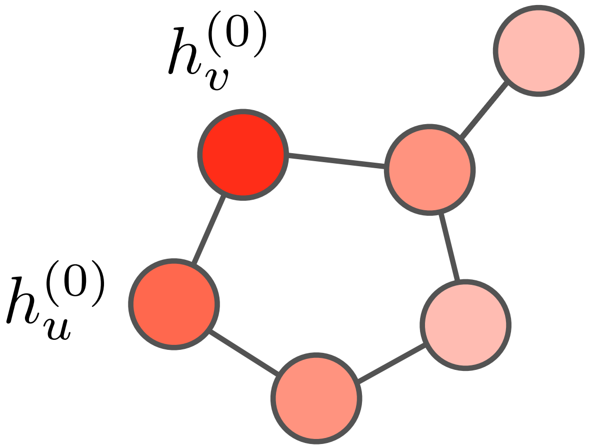
➡ 一般の GNN は計算が重く、大きいグラフ（~ 1M 節点）に使用できない

GCN: 軽量な GNN

- GCN: Graph Convolutional Network (Kipf+ 2017)
  - ノード間の相互作用を線形化しモデルを軽量化

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}(A + I)\tilde{D}^{-\frac{1}{2}}$$

$$h_v^{(k)} = \mathrm{Activation}^{(k)}\left(\sum_u \hat{A}_{vu} h_u^{(k-1)} W^{(k)}\right)$$

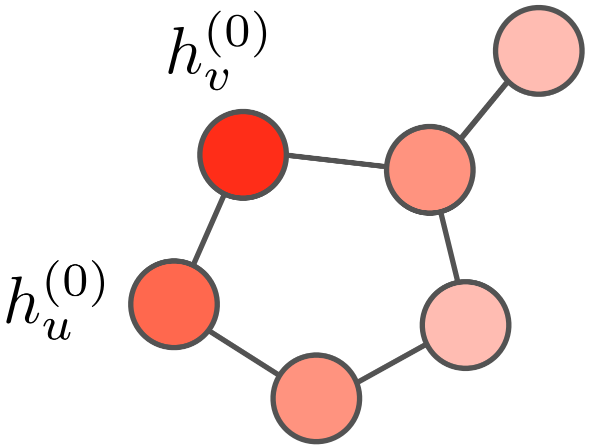


GCN: 軽量な GNN

- GCN: Graph Convolutional Network (Kipf+ 2017)
  - ノード間の相互作用を線形化しモデルを軽量化

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}(A+I)\tilde{D}^{-\frac{1}{2}}$$

$$h_v^{(k)} = \mathrm{Activation}^{(k)}\left(\sum_u \hat{A}_{vu} h_u^{(k-1)} W^{(k)}\right)$$



$|\mathcal{V}|$: グラフのノード数

$A \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$: 隣接行列

$\tilde{D}$: $A+I$ の度数行列

$h^{(k)} \in \mathbb{R}^{|\mathcal{V}|\times f^{(k)}}$: 特徴量行列

$W^{(k)} \in \mathbb{R}^{f^{(k-1)}\times f^{(k)}}$:
学習するパラメータ行列

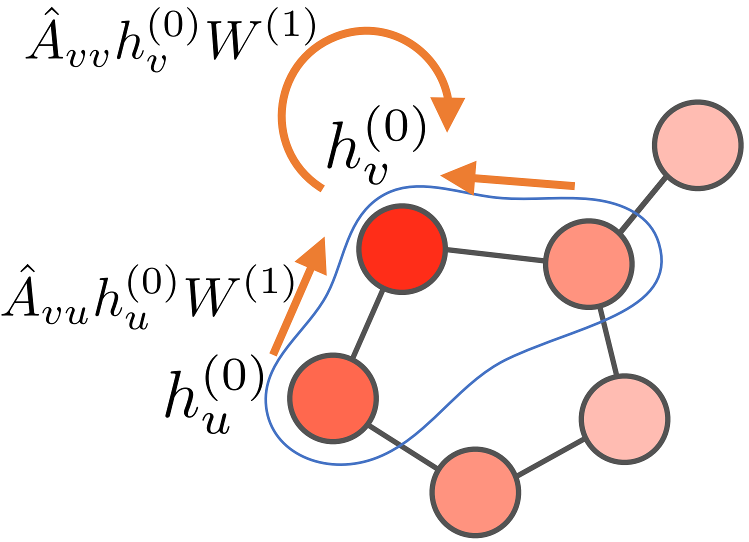
$\mathrm{Activation}^{(k)}$:
成分ごとに作用する非線形関数

GCN: 軽量な GNN

- GCN: Graph Convolutional Network (Kipf+ 2017)
  - ノード間の相互作用を線形化しモデルを軽量化

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}(A + I)\tilde{D}^{-\frac{1}{2}}$$

$$h_v^{(k)} = \mathrm{Activation}^{(k)}\left(\sum_u \hat{A}_{vu} h_u^{(k-1)} W^{(k)}\right)$$



$|\mathcal{V}|$: グラフのノード数

$A \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$: 隣接行列

$\tilde{D}$: $A + I$ の度数行列

$h^{(k)} \in \mathbb{R}^{|\mathcal{V}| \times f^{(k)}}$: 特徴量行列

$W^{(k)} \in \mathbb{R}^{f^{(k-1)} \times f^{(k)}}$:
学習するパラメータ行列

$\mathrm{Activation}^{(k)}$:
成分ごとに作用する非線形関数

GCN: 軽量な GNN

- GCN: Graph Convolutional Network (Kipf+ 2017)
  - ノード間の相互作用を線形化しモデルを軽量化

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}(A+I)\tilde{D}^{-\frac{1}{2}}$$

$$h_v^{(k)} = \mathrm{Activation}^{(k)}\left(\sum_u \hat{A}_{vu} h_u^{(k-1)} W^{(k)}\right)$$

$|\mathcal{V}|$: グラフのノード数

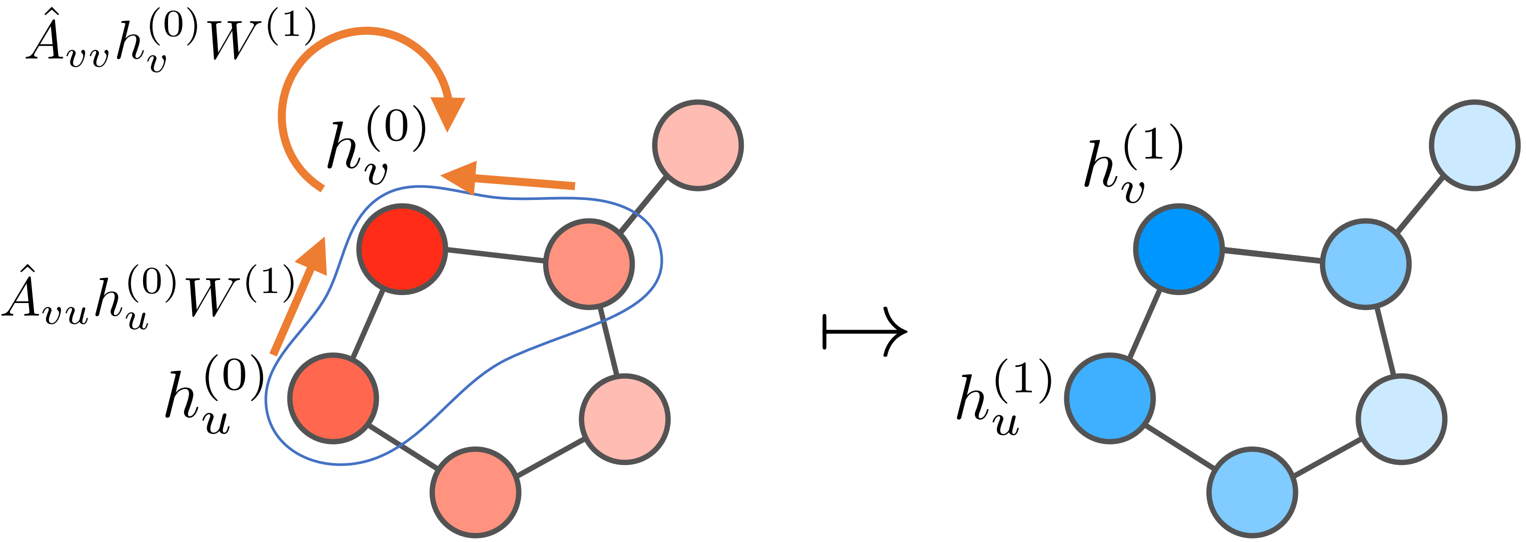
$A \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$: 隣接行列

$\tilde{D}$: $A+I$ の度数行列

$h^{(k)} \in \mathbb{R}^{|\mathcal{V}|\times f^{(k)}}$: 特徴量行列

$W^{(k)} \in \mathbb{R}^{f^{(k-1)}\times f^{(k)}}$: 学習するパラメータ行列

$\mathrm{Activation}^{(k)}$: 成分ごとに作用する非線形関数



GCN: 軽量な GNN

- GCN: Graph Convolutional Network (Kipf+ 2017)
  - ノード間の相互作用を線形化しモデルを軽量化

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}(A + I)\tilde{D}^{-\frac{1}{2}}$$

$$h_v^{(k)} = \mathrm{Activation}^{(k)}\left(\sum_u \hat{A}_{vu} h_u^{(k-1)} W^{(k)}\right)$$

$|\mathcal{V}|$: グラフのノード数

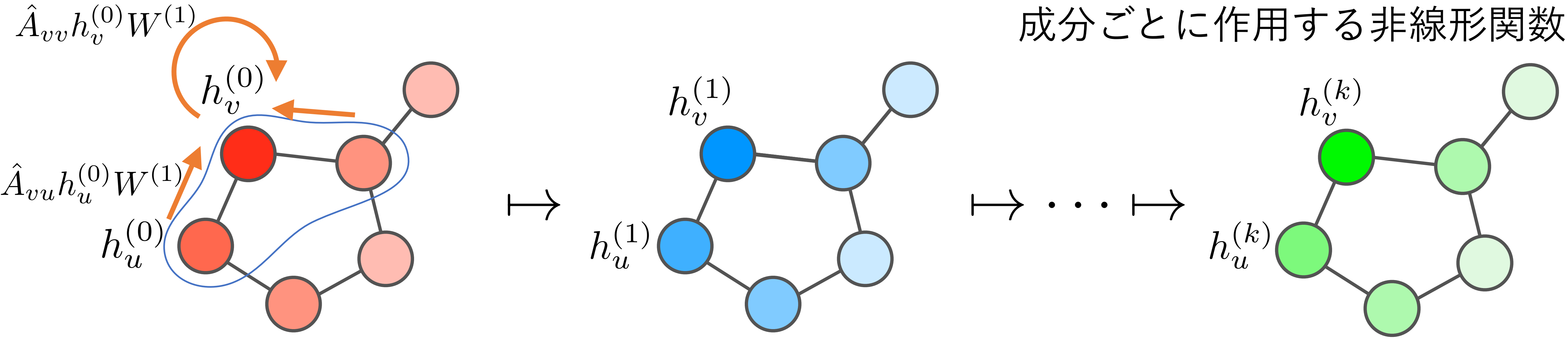
$A \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$: 隣接行列

$\tilde{D}$: $A + I$ の度数行列

$h^{(k)} \in \mathbb{R}^{|\mathcal{V}| \times f^{(k)}}$: 特徴量行列

$W^{(k)} \in \mathbb{R}^{f^{(k-1)} \times f^{(k)}}$:
学習するパラメータ行列

$\mathrm{Activation}^{(k)}$:
成分ごとに作用する非線形関数



目次

目次

不変性と同変性：現象の持つ対称性

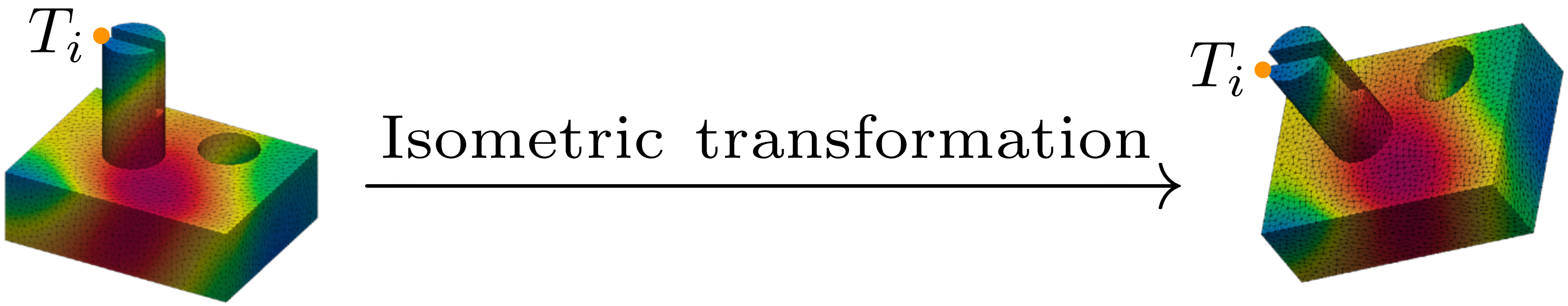
G を群、X、Y を左 G-集合とする

- 関数 f: X → Y が群 G の作用に対して**不変**（invariant）である：

$$\forall g \in G, \forall x \in X, f(g \cdot x) = f(x)$$

$$\begin{tikzcd} X \arrow[r, "g\cdot"] \arrow[dr, "f"'] & X \arrow[d, "f"] \\ & Y \end{tikzcd}$$

- 例：温度は合同変換（回転・鏡映・平行移動）に対して不変



不変性と同変性：現象の持つ対称性

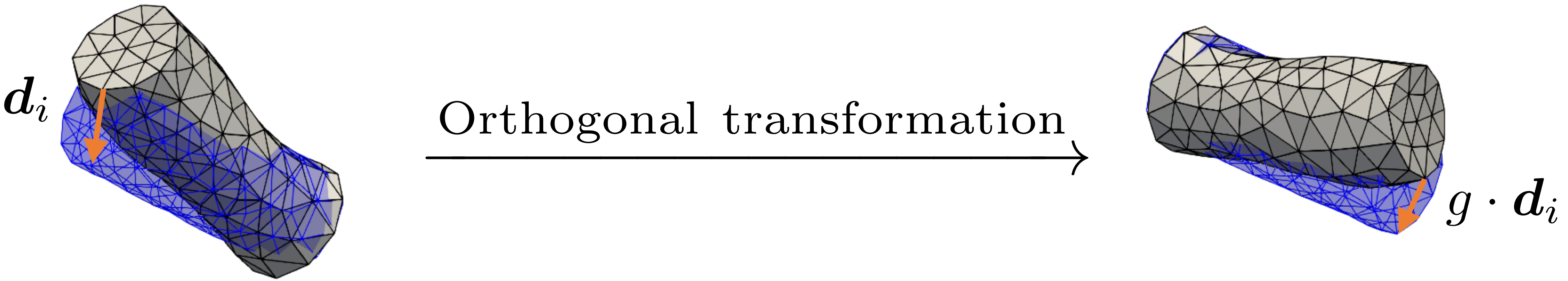
G を群、X、Y を左 G-集合とする

- 関数 f: X → Y が群 G の作用に対して**同変**（equivariant）である：

$$\forall g \in G, \forall x \in X, f(g \cdot x) = g \cdot f(x)$$

$$\begin{tikzcd} X \arrow[r, "g\cdot"] \arrow[d, "f"] & X \arrow[d, "f"] \\ Y \arrow[r, "g\cdot"] & Y \end{tikzcd}$$

- 例：変位は直交変換（回転・鏡映）に対して同変
  - 変位は平行移動に対して不変
  - 不変性は同変性の一種なので、変位は合同変換に対して同変

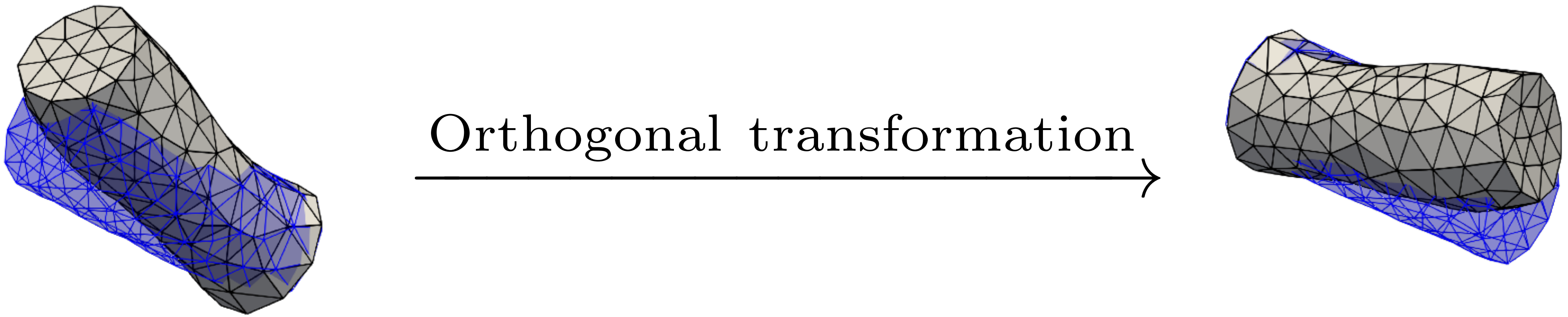


不変性と同変性：現象の持つ対称性

合同変換の群（ユークリッド群）を $E(n)$ とする

- 物理量・物理法則は合同変換同変である
  - 例：フックの法則

$$\forall e \in E(n), \sigma = -C\varepsilon \Rightarrow e \cdot \sigma = -(e \cdot C)(e \cdot \varepsilon)$$



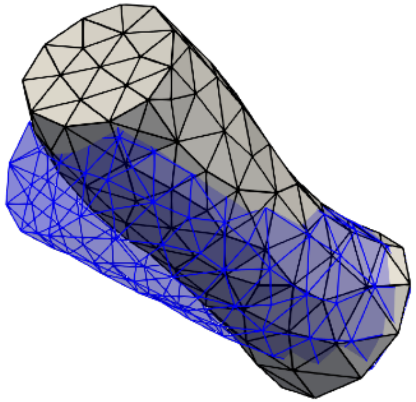
不変性と同変性：現象の持つ対称性

合同変換の群（ユークリッド群）を $E(n)$ とする

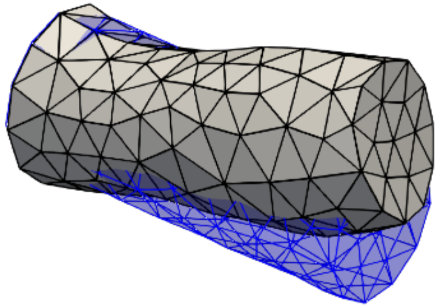
- 物理量・物理法則は合同変換同変である
  - 例：フックの法則

$$\forall e \in E(n), \sigma = -C\varepsilon \Rightarrow e \cdot \sigma = -(e \cdot C)(e \cdot \varepsilon)$$

$$\begin{aligned}
U^e \in O(n), U^e_{ii'} U^e_{jj'} \sigma_{i'j'} &= -U^e_{ii'} U^e_{jj'} U^e_{kk'} U^e_{ll'} C_{i'j'k'l'} U^e_{kk''} U^e_{ll''} \varepsilon_{k''l''} \\
&= -U^e_{ii'} U^e_{jj'} \delta_{k'k''} \delta_{l'l''} C_{i'j'k'l'} \varepsilon_{k''l''} \\
&= -U^e_{ii'} U^e_{jj'} C_{i'j'k'l'} \varepsilon_{k'l'}
\end{aligned}$$



$$\xrightarrow{\text{Orthogonal transformation}}$$



GCN と不変性・同変性

- GCN: Graph Convolutional Network (Kipf+ 2017)
  - ノード間の相互作用を線形化しモデルを軽量化

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}(A+I)\tilde{D}^{-\frac{1}{2}}$$

$$h_v^{(k)} = \mathrm{Activation}^{(k)}\left(\sum_u \hat{A}_{vu} h_u^{(k-1)} W^{(k)}\right)$$

$|\mathcal{V}|$: グラフのノード数

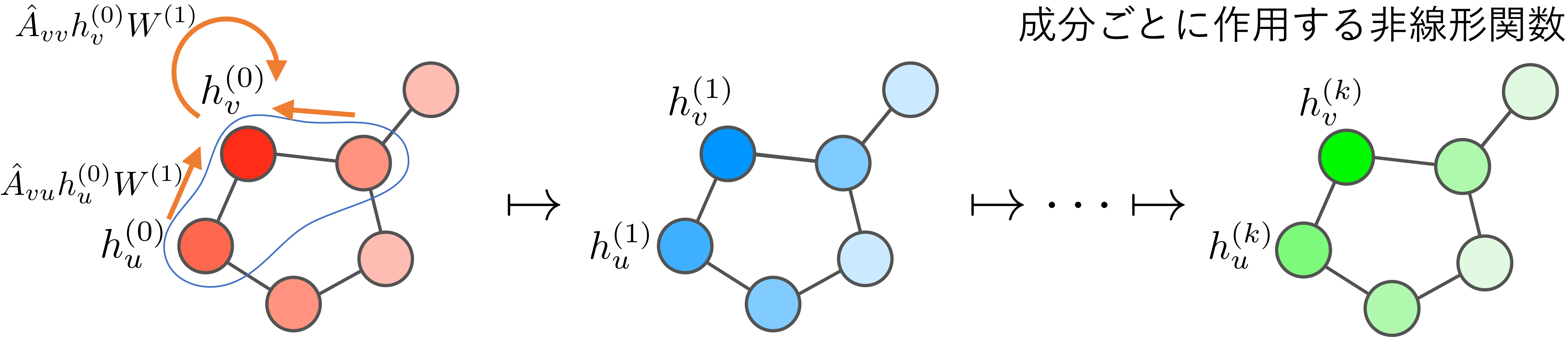
$A \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$: 隣接行列

$\tilde{D}$: $A+I$ の度数行列

$h^{(k)} \in \mathbb{R}^{|\mathcal{V}|\times f^{(k)}}$: 特徴量行列

$W^{(k)} \in \mathbb{R}^{f^{(k-1)}\times f^{(k)}}$:
学習するパラメータ行列

$\mathrm{Activation}^{(k)}$:
成分ごとに作用する非線形関数



GCN と不変性・同変性

- GCN: Graph Convolutional Network (Kipf+ 2017)
  - ノード間の相互作用を線形化しモデルを軽量化

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}(A+I)\tilde{D}^{-\frac{1}{2}}$$

$$h_v^{(k)} = \mathrm{Activation}^{(k)}\left(\sum_u \hat{A}_{vu} h_u^{(k-1)} W^{(k)}\right)$$

- GCN は、入力が合同変換不変のときのみ不変
  - 不変性を保ったままノード位置を入力できない

$|\mathcal{V}|$: グラフのノード数

$A \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$: 隣接行列

$\tilde{D}$: $A+I$ の度数行列

$h^{(k)} \in \mathbb{R}^{|\mathcal{V}|\times f^{(k)}}$: 特徴量行列

$W^{(k)} \in \mathbb{R}^{f^{(k-1)}\times f^{(k)}}$:
学習するパラメータ行列

$\mathrm{Activation}^{(k)}$:
成分ごとに作用する非線形関数

GCN と不変性・同変性

- GCN: Graph Convolutional Network (Kipf+ 2017)
  - ノード間の相互作用を線形化しモデルを軽量化

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}(A+I)\tilde{D}^{-\frac{1}{2}}$$

$$h_v^{(k)} = \mathrm{Activation}^{(k)}\left(\sum_u \hat{A}_{vu} h_u^{(k-1)} W^{(k)}\right)$$

- GCN は、入力が合同変換不変のときのみ不変
  - 不変性を保ったままノード位置を入力できない

$|\mathcal{V}|$: グラフのノード数

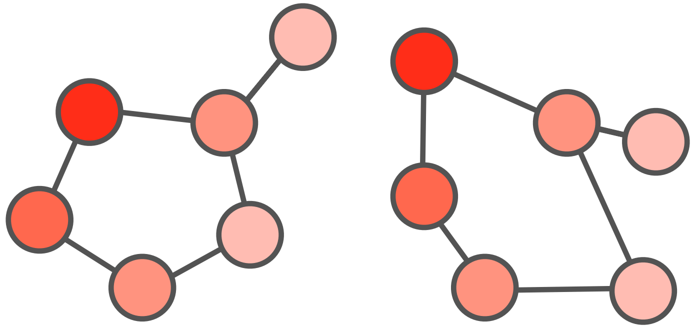
$A \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$: 隣接行列

$\tilde{D}$: $A+I$ の度数行列

$h^{(k)} \in \mathbb{R}^{|\mathcal{V}|\times f^{(k)}}$: 特徴量行列

$W^{(k)} \in \mathbb{R}^{f^{(k-1)}\times f^{(k)}}$:
学習するパラメータ行列

$\mathrm{Activation}^{(k)}$:
成分ごとに作用する非線形関数



GCN と不変性・同変性

- GCN: Graph Convolutional Network (Kipf+ 2017)
  - ノード間の相互作用を線形化しモデルを軽量化

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}(A+I)\tilde{D}^{-\frac{1}{2}}$$

$$h_v^{(k)} = \mathrm{Activation}^{(k)}\left(\sum_u \hat{A}_{vu} h_u^{(k-1)} W^{(k)}\right)$$

- GCN は、入力が合同変換不変のときのみ不変
  - 不変性を保ったままノード位置を入力できない
- 一般に GCN は合同変換同変とならない

$$\exists e \in E(d), e \cdot h^{(k)} \neq \mathrm{Activation}^{(k)}(\hat{A}e \cdot h^{(k-1)} W^{(k)})$$

$|\mathcal{V}|$: グラフのノード数

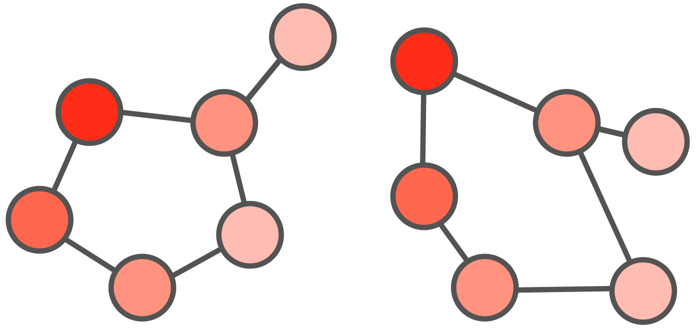
$A \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$: 隣接行列

$\tilde{D}$: $A+I$ の度数行列

$h^{(k)} \in \mathbb{R}^{|\mathcal{V}|\times f^{(k)}}$: 特徴量行列

$W^{(k)} \in \mathbb{R}^{f^{(k-1)}\times f^{(k)}}$:
学習するパラメータ行列

$\mathrm{Activation}^{(k)}$:
成分ごとに作用する非線形関数

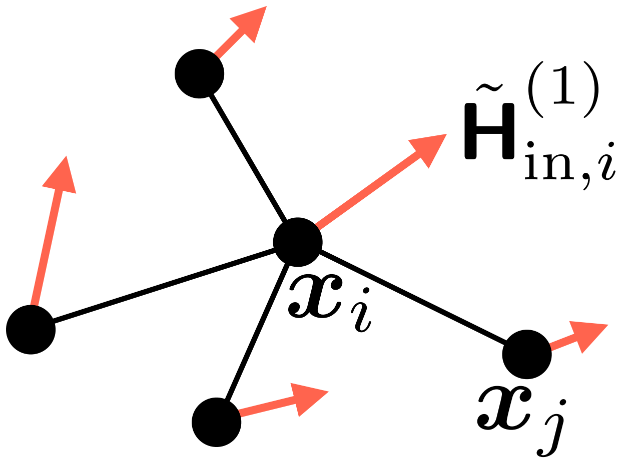


Tensor field network: 合同変換同変な機械学習モデル

- TFN: Tensor field network (Thomas+ 2018)
  - 球面調和関数を用いた
    合同変換不変・同変な機械学習モデル

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

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

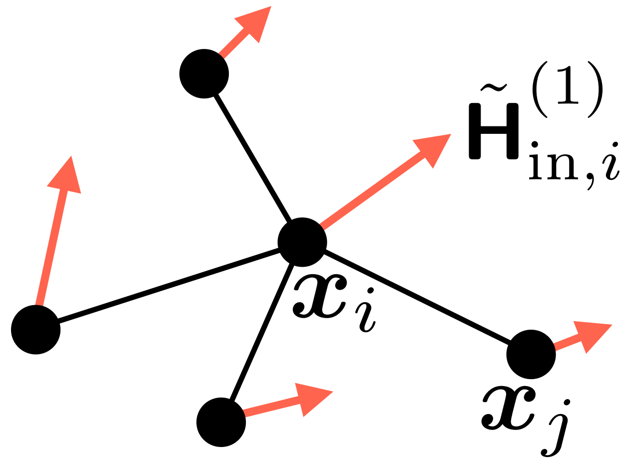


Tensor field network: 合同変換同変な機械学習モデル

- TFN: Tensor field network (Thomas+ 2018)
  - 球面調和関数を用いた合同変換不変・同変な機械学習モデル

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

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



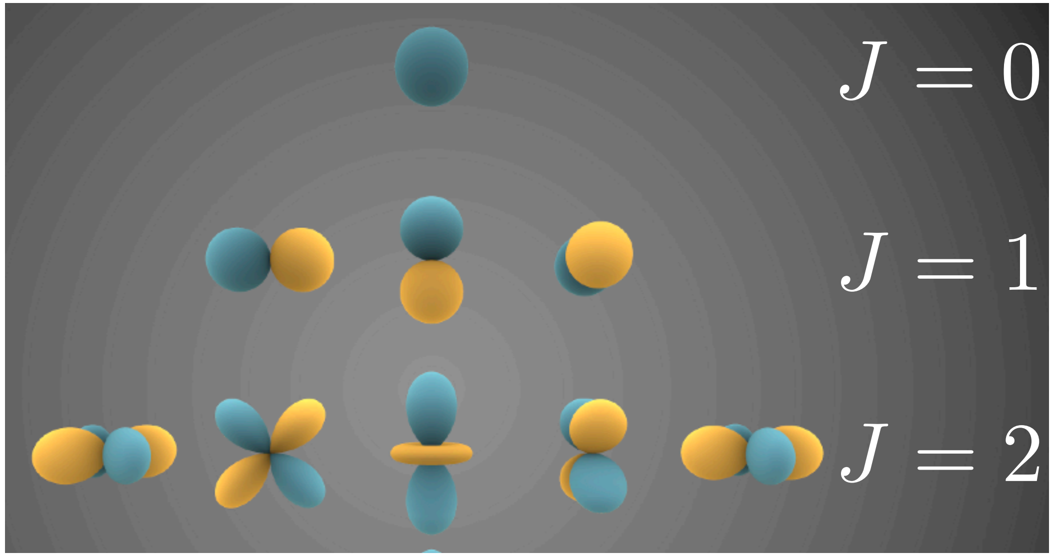
$\tilde{\mathbf{H}}^{(l)}_{\mathrm{in},i} \in \mathbb{R}^{2l+1}$: type-l 入力（l階テンソルに対応）

$\boldsymbol{x}_i$: i 番目ノード位置

w^{ll} : 学習するパラメータ

$\phi^{lk}_J : \mathbb{R}_{\geq 0} \to \mathbb{R}$: 学習する関数

$Y_J : \mathbb{R}^3 \to \mathbb{R}^{2J+1}$:球面調和関数



Tensor field network: 合同変換同変な機械学習モデル

- TFN: Tensor field network (Thomas+ 2018)
  - 球面調和関数を用いた
    合同変換不変・同変な機械学習モデル

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

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

$\tilde{\mathbf{H}}^{(l)}_{\mathrm{in},i} \in \mathbb{R}^{2l+1}$: type-l 入力（l階テンソルに対応）

$\boldsymbol{x}_i$: i 番目ノード位置

w^{ll} : 学習するパラメータ

$\phi^{lk}_J : \mathbb{R}_{\geq 0} \to \mathbb{R}$: 学習する関数

$Y_J : \mathbb{R}^3 \to \mathbb{R}^{2J+1}$:球面調和関数

$\boldsymbol{Q}^{lk}_{Jm} \in \mathbb{R}^{(2l+1)\times(2k+1)}$: Clebsch-Gordan 係数

Tensor field network: 合同変換同変な機械学習モデル

- TFN: Tensor field network (Thomas+ 2018)
  - 球面調和関数を用いた
    合同変換不変・同変な機械学習モデル

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

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

$\tilde{\mathbf{H}}^{(l)}_{\mathrm{in},i} \in \mathbb{R}^{2l+1}$: type-l 入力

（l階テンソルに対応）

$\boldsymbol{x}_i$: i番目ノード位置

w^{ll} : 学習するパラメータ

$\phi^{lk}_J : \mathbb{R}_{\geq 0} \to \mathbb{R}$: 学習する関数

$Y_J : \mathbb{R}^3 \to \mathbb{R}^{2J+1}$:球面調和関数

$\boldsymbol{Q}^{lk}_{Jm} \in \mathbb{R}^{(2l+1)\times(2k+1)}$: Clebsch-Gordan 係数

type-Jと type-k の入力から type-l の出力を返すときの係数

(J,k,l)	$(0,0,0)$	$(0,1,1)$ $(1,0,1)$	$(1,1,0)$	$(1,1,1)$
対応物	掛け算	ベクトルのスカラー倍	ドット積	クロス積

Tensor field network: 合同変換同変な機械学習モデル

- TFN: Tensor field network (Thomas+ 2018)
  - 球面調和関数を用いた
    合同変換不変・同変な機械学習モデル

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

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

$\tilde{\mathbf{H}}^{(l)}_{\mathrm{in},i} \in \mathbb{R}^{2l+1}$: type-l 入力（l階テンソルに対応）

$\boldsymbol{x}_i$: i 番目ノード位置

w^{ll} : 学習するパラメータ

$\phi^{lk}_J : \mathbb{R}_{\geq 0} \to \mathbb{R}$: 学習する関数

$Y_J : \mathbb{R}^3 \to \mathbb{R}^{2J+1}$:球面調和関数

$\boldsymbol{Q}^{lk}_{Jm} \in \mathbb{R}^{(2l+1)\times(2k+1)}$: Clebsch-Gordan 係数

- Universality （同変な連続関数を任意の精度で一様に近似できる性質）が示されている（Dym+ 2020）
- 計算に時間がかかるためシミュレーションの代替には不向き

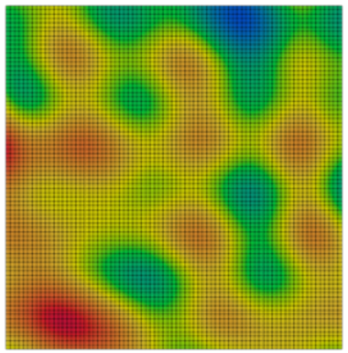
目次

- Graph neural network
- 同変性
- IsoGCN: 同変性＋Graph Neural Network
- まとめ

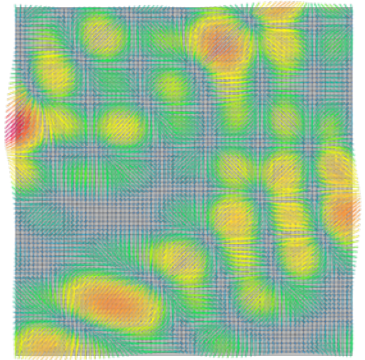
目次

概要：対称性があり計算効率のよい GNN を開発

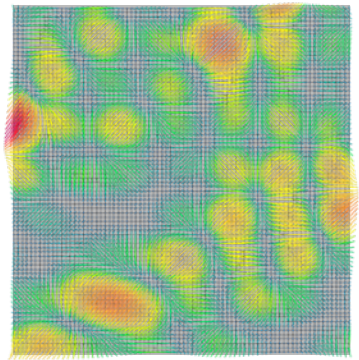
- IsoGCN: 合同変換不変・同変で計算効率のよい GNN (Horie+ 2020)
- IsoGCN は同等の精度の有限要素解析より高速
- IsoGCN は微分演算子の近似とみなせる



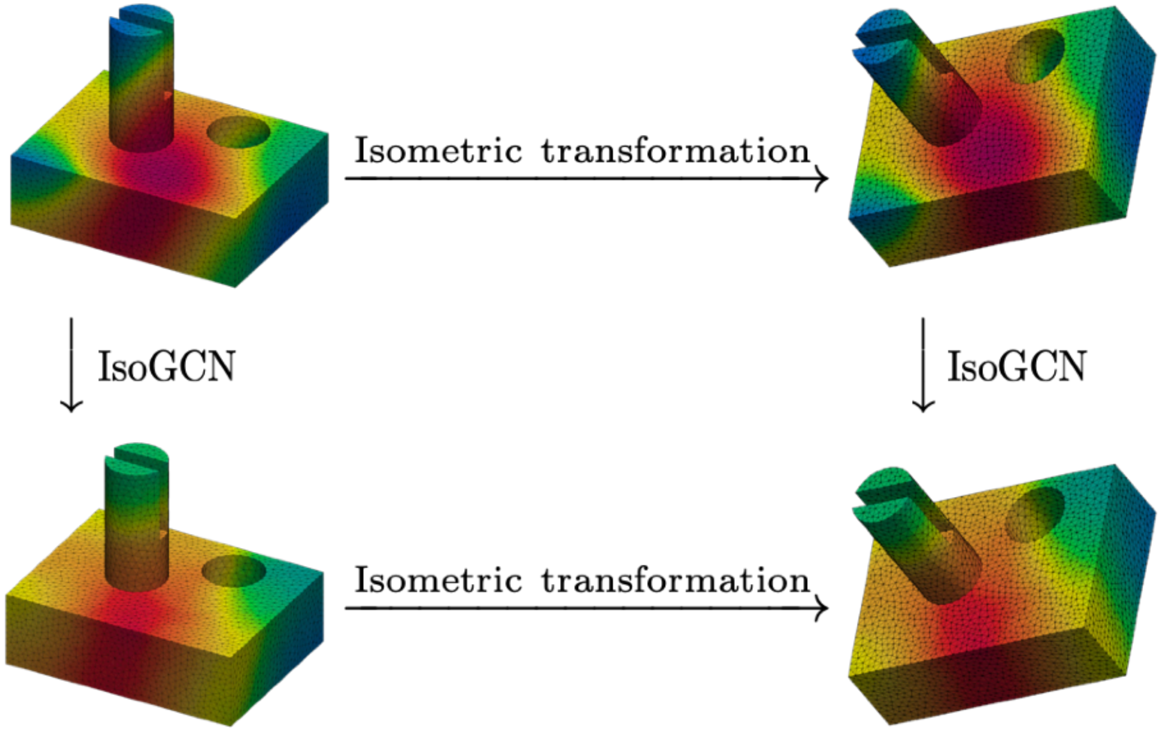
スカラー場
（入力）



勾配
（正解）



勾配
（推論）



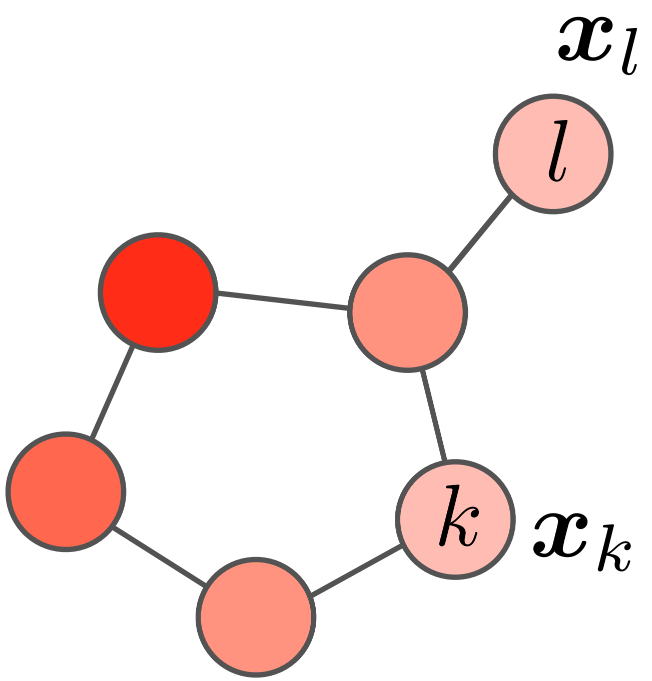
IsoGCN：合同変換不変・同変な GCN

頂点数$|\mathcal{V}|$ のグラフが d 次元ユークリッド空間に
埋め込まれているとする

- IsoAM (Isometric Adjacency Matrix) G：

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

$$G_{ijm} = \sum_{k,l;k \neq l} \left[\boldsymbol{T}_{ijkl}(\boldsymbol{x}_k - \boldsymbol{x}_l) \right]_m$$



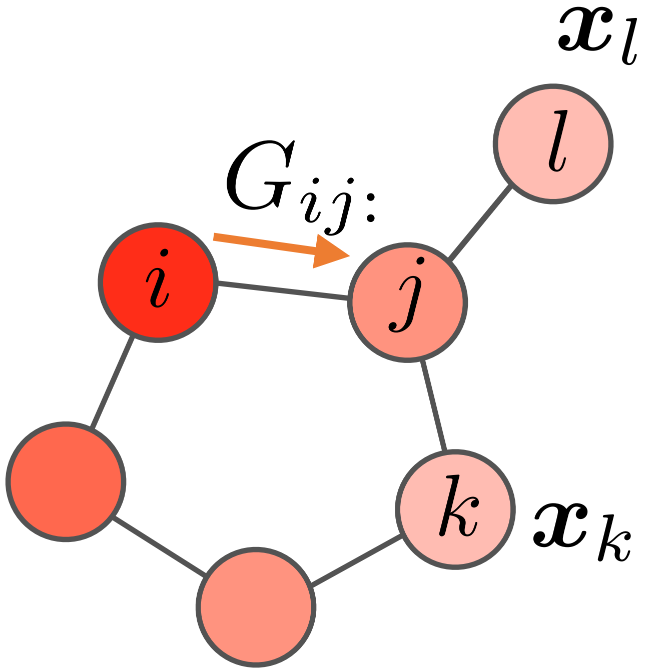
IsoGCN：合同変換不変・同変な GCN

頂点数 $|\mathcal{V}|$ のグラフが d 次元ユークリッド空間に埋め込まれているとする

- IsoAM (Isometric Adjacency Matrix) G：

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

$$G_{ijm} = \sum_{k,l;k \neq l} \left[\boldsymbol{T}_{ijkl} (\boldsymbol{x}_k - \boldsymbol{x}_l) \right]_m$$



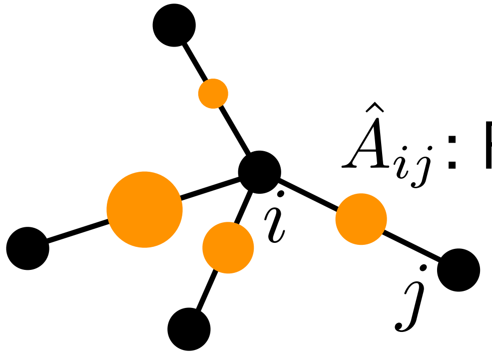
- $\boldsymbol{T}_{ijkl}$：グラフごとに定まる 2 階テンソル
  - $\boldsymbol{T}_{ijkl} = \delta_{il}\delta_{jk}\boldsymbol{I}$ とおけば

$$G_{ij:} = \boldsymbol{x}_j - \boldsymbol{x}_i$$

Key Idea: GCN with Vector-Valued Adjacency Matrix

- GCN's message passing

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

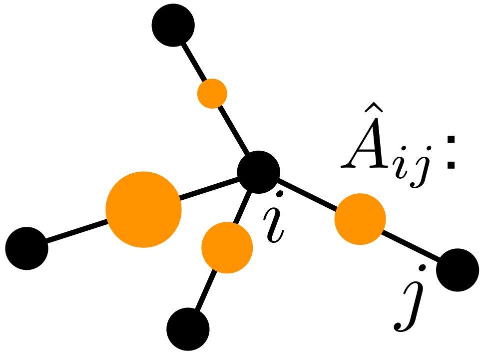


$\hat{A}_{ij}$: Renormalized adjacency matrix

Key Idea: GCN with Vector-Valued Adjacency Matrix

- GCN's message passing

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

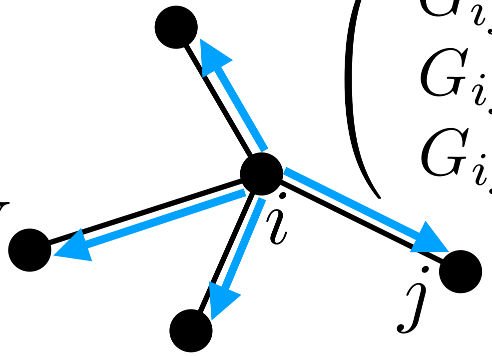


$\hat{A}_{ij}$: Renormalized adjacency matrix

- IsoGCN's message passing:

  - Scalar to vector

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



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

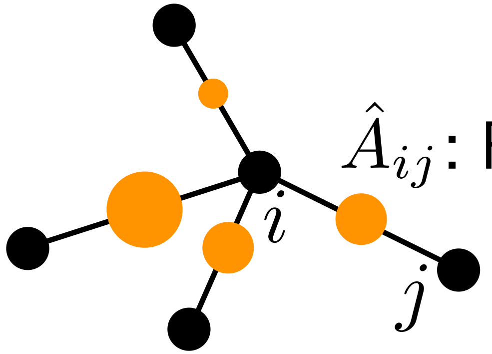
IsoAM :

relative vertex positions

Key Idea: GCN with Vector-Valued Adjacency Matrix

- GCN's message passing

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

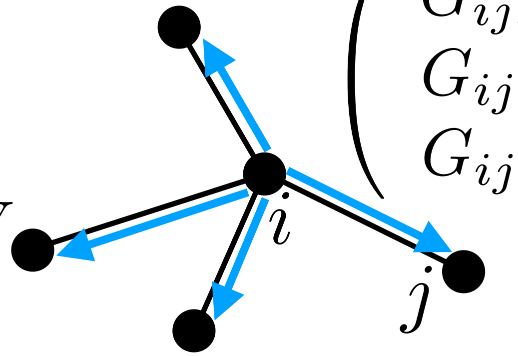


$\hat{A}_{ij}$: Renormalized adjacency matrix

- IsoGCN's message passing:

  - Scalar to vector

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



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

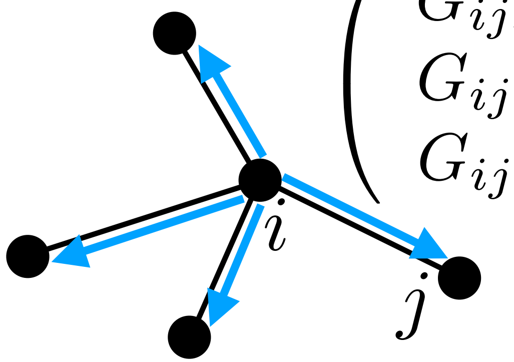
IsoAM :

relative vertex positions

Key Idea: GCN with Vector-Valued Adjacency Matrix

- GCN'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

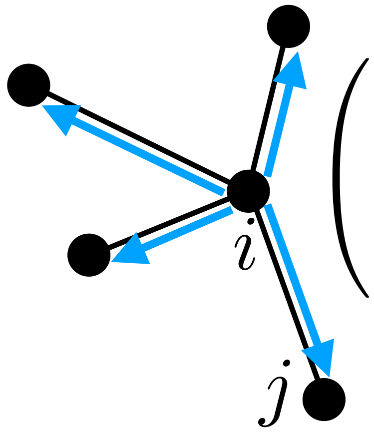
$$\begin{pmatrix} H_{\mathrm{out},x} \\ H_{\mathrm{out},y} \\ H_{\mathrm{out},z} \end{pmatrix} = \boxed{\begin{pmatrix} G_{::,x} \\ G_{::,y} \\ G_{::,z} \end{pmatrix}} H_{\mathrm{in}} W$$

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

Key Idea: GCN with Vector-Valued Adjacency Matrix

- GCN'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

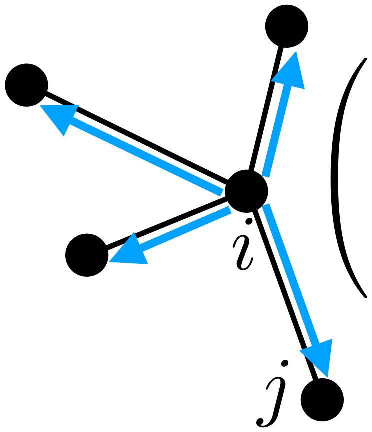
$$\begin{pmatrix} H_{\mathrm{out},x} \\ H_{\mathrm{out},y} \\ H_{\mathrm{out},z} \end{pmatrix} = \boxed{\begin{pmatrix} G_{::,x} \\ G_{::,y} \\ G_{::,z} \end{pmatrix}} H_{\mathrm{in}} W$$

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

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

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

IsoGCN：合同変換不変・同変な GCN

- 有限要素解析に適用するため
  IsoAM の具体例 $\tilde{D}_{ijk}$ を以下のように定義する

$$\tilde{D}_{ijk} = D_{ijk} - \delta_{ij} \sum_{l} D_{ilk}$$

$$D_{ijk} = M^{-1} \frac{x_{jk} - x_{ik}}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2} \frac{V_j^{\text{effective}}}{V_i^{\text{effective}}} A_{ij}$$

$$\boldsymbol{M}_i = \sum_{l} \frac{\boldsymbol{x}_l - \boldsymbol{x}_i}{\|\boldsymbol{x}_l - \boldsymbol{x}_i\|} \otimes \frac{\boldsymbol{x}_l - \boldsymbol{x}_i}{\|\boldsymbol{x}_l - \boldsymbol{x}_i\|} \frac{V_j^{\text{effective}}}{V_i^{\text{effective}}} A_{il}$$

IsoGCN：合同変換不変・同変な GCN

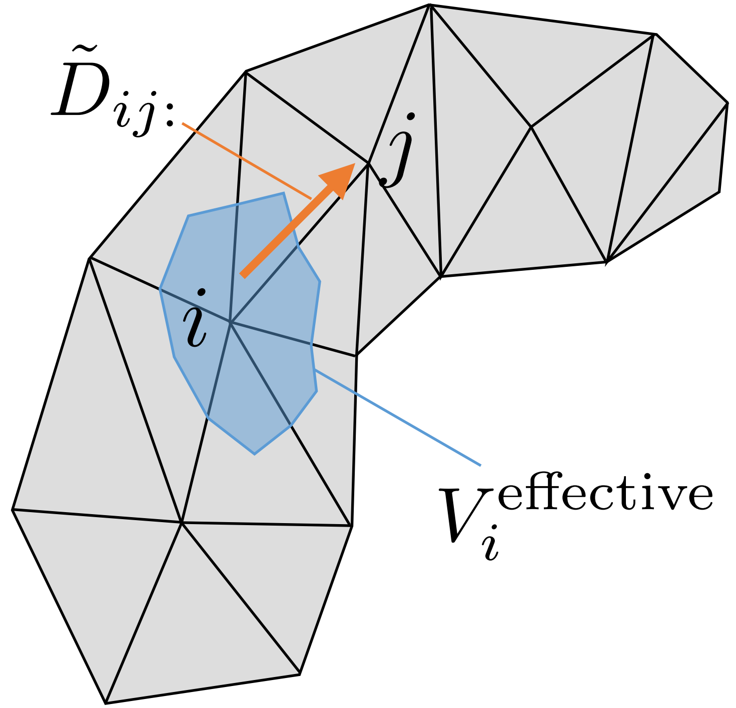
- 有限要素解析に適用するため
  IsoAM の具体例 $\tilde{D}_{ijk}$ を以下のように定義する

$$\tilde{D}_{ijk} = D_{ijk} - \delta_{ij} \sum_l D_{ilk}$$

$$D_{ijk} = M^{-1} \frac{x_{jk} - x_{ik}}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2} \frac{V_j^{\text{effective}}}{V_i^{\text{effective}}} A_{ij}$$

$$\boldsymbol{M}_i = \sum_l \frac{\boldsymbol{x}_l - \boldsymbol{x}_i}{\|\boldsymbol{x}_l - \boldsymbol{x}_i\|} \otimes \frac{\boldsymbol{x}_l - \boldsymbol{x}_i}{\|\boldsymbol{x}_l - \boldsymbol{x}_i\|} \frac{V_j^{\text{effective}}}{V_i^{\text{effective}}} A_{il}$$

A ：隣接行列
$V_i^{\text{effective}}$：頂点 i の体積



IsoGCN：合同変換不変・同変な GCN

- 有限要素解析に適用するため
  IsoAM の具体例 $\tilde{D}_{ijk}$ を以下のように定義する

$$\tilde{D}_{ijk} = D_{ijk} - \delta_{ij} \sum_l D_{ilk}$$

$$D_{ijk} = M^{-1} \frac{x_{jk} - x_{ik}}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2} \frac{V_j^{\text{effective}}}{V_i^{\text{effective}}} A_{ij}$$

$$\boldsymbol{M}_i = \sum_l \frac{\boldsymbol{x}_l - \boldsymbol{x}_i}{\|\boldsymbol{x}_l - \boldsymbol{x}_i\|} \otimes \frac{\boldsymbol{x}_l - \boldsymbol{x}_i}{\|\boldsymbol{x}_l - \boldsymbol{x}_i\|} \frac{V_j^{\text{effective}}}{V_i^{\text{effective}}} A_{il}$$

- $\tilde{D}$ がナブラ演算子の離散化（Tamai+ 2014）に対応している

$$\left\langle \frac{\partial \phi}{\partial x_k} \right\rangle_i := \boldsymbol{M}_i^{-1} \sum_j \frac{\phi_j - \phi_i}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|} \frac{x_{jk} - x_{ik}}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|} \frac{V_j^{\text{effective}}}{V_i^{\text{effective}}} A_{ij} = \sum_j \tilde{D}_{ijk} \phi_j$$

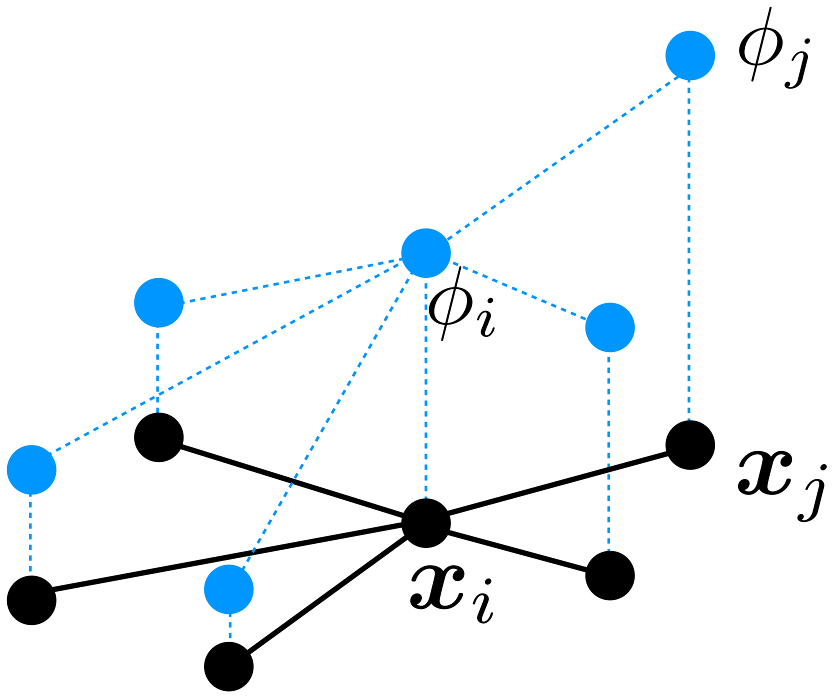
IsoGCN：合同変換不変・同変な GCN

- 有限要素解析に適用するため
  IsoAM の具体例 $\tilde{D}_{ijk}$ を以下のように定義する

$$\tilde{D}_{ijk} = D_{ijk} - \delta_{ij} \sum_{l} D_{ilk}$$

$$D_{ijk} = M^{-1} \frac{x_{jk} - x_{ik}}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2} \frac{V_j^{\text{effective}}}{V_i^{\text{effective}}} A_{ij}$$

$$\boldsymbol{M}_i = \sum_{l} \frac{\boldsymbol{x}_l - \boldsymbol{x}_i}{\|\boldsymbol{x}_l - \boldsymbol{x}_i\|} \otimes \frac{\boldsymbol{x}_l - \boldsymbol{x}_i}{\|\boldsymbol{x}_l - \boldsymbol{x}_i\|} \frac{V_j^{\text{effective}}}{V_i^{\text{effective}}} A_{il}$$



- $\tilde{D}$ がナブラ演算子の離散化（Tamai+ 2014）に対応している

$$\left\langle \frac{\partial \phi}{\partial x_k} \right\rangle_i := \boldsymbol{M}_i^{-1} \sum_{j} \frac{\phi_j - \phi_i}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|} \frac{x_{jk} - x_{ik}}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|} \frac{V_j^{\text{effective}}}{V_i^{\text{effective}}} A_{ij} = \sum_{j} \tilde{D}_{ijk} \phi_j$$

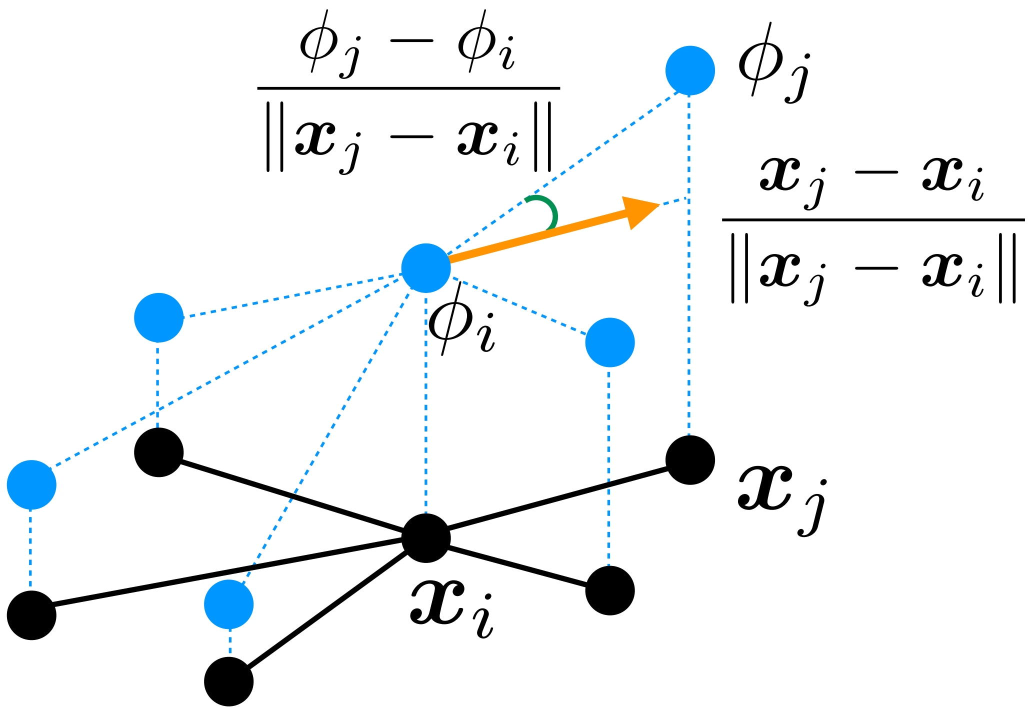
IsoGCN：合同変換不変・同変な GCN

- 有限要素解析に適用するため
  IsoAM の具体例 $\tilde{D}_{ijk}$ を以下のように定義する

$$\tilde{D}_{ijk} = D_{ijk} - \delta_{ij} \sum_l D_{ilk}$$

$$D_{ijk} = M^{-1} \frac{x_{jk} - x_{ik}}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2} \frac{V_j^{\text{effective}}}{V_i^{\text{effective}}} A_{ij}$$

$$\boldsymbol{M}_i = \sum_l \frac{\boldsymbol{x}_l - \boldsymbol{x}_i}{\|\boldsymbol{x}_l - \boldsymbol{x}_i\|} \otimes \frac{\boldsymbol{x}_l - \boldsymbol{x}_i}{\|\boldsymbol{x}_l - \boldsymbol{x}_i\|} \frac{V_j^{\text{effective}}}{V_i^{\text{effective}}} A_{il}$$



- $\tilde{D}$ がナブラ演算子の離散化（Tamai+ 2014）に対応している

$$\left\langle \frac{\partial \phi}{\partial x_k} \right\rangle_i := \boldsymbol{M}_i^{-1} \sum_j \frac{\phi_j - \phi_i}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|} \frac{x_{jk} - x_{ik}}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|} \frac{V_j^{\text{effective}}}{V_i^{\text{effective}}} A_{ij} = \sum_j \tilde{D}_{ijk} \phi_j$$

IsoGCN：合同変換不変・同変な GCN

- 有限要素解析に適用するため
IsoAM の具体例 $\tilde{D}_{ijk}$ を以下のように定義する

$$\tilde{D}_{ijk} = D_{ijk} - \delta_{ij} \sum_{l} D_{ilk}$$

$$D_{ijk} = M^{-1} \frac{x_{jk} - x_{ik}}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2} \frac{V_j^{\text{effective}}}{V_i^{\text{effective}}} A_{ij}$$

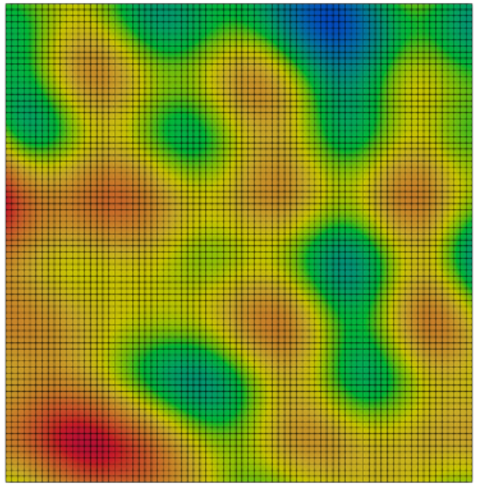
$$\boldsymbol{M}_i = \sum_{l} \frac{\boldsymbol{x}_l - \boldsymbol{x}_i}{\|\boldsymbol{x}_l - \boldsymbol{x}_i\|} \otimes \frac{\boldsymbol{x}_l - \boldsymbol{x}_i}{\|\boldsymbol{x}_l - \boldsymbol{x}_i\|} \frac{V_j^{\text{effective}}}{V_i^{\text{effective}}} A_{il}$$

Differential op.	**Expression**
Gradient	$\tilde{\mathbf{D}} * \mathbf{H}^{(0)}$
Divergence	$\tilde{\mathbf{D}} \odot \mathbf{H}^{(1)}$
Laplacian	$\tilde{\mathbf{D}} \odot \tilde{\mathbf{D}}\mathbf{H}^{(0)}$
Jacobian	$\tilde{\mathbf{D}} \otimes \mathbf{H}^{(1)}$
Hessian	$\tilde{\mathbf{D}} \otimes \tilde{\mathbf{D}} * \mathbf{H}^{(0)}$

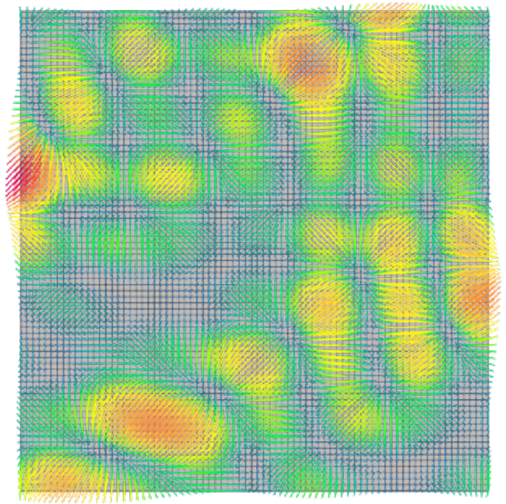
- $\tilde{D}$ がナブラ演算子の離散化（Tamai+ 2014）に対応している

$$\left\langle \frac{\partial \phi}{\partial x_k} \right\rangle_i := \boldsymbol{M}_i^{-1} \sum_j \frac{\phi_j - \phi_i}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|} \frac{x_{jk} - x_{ik}}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|} \frac{V_j^{\text{effective}}}{V_i^{\text{effective}}} A_{ij} = \sum_j \tilde{D}_{ijk} \phi_j$$

実験：微分演算子の学習



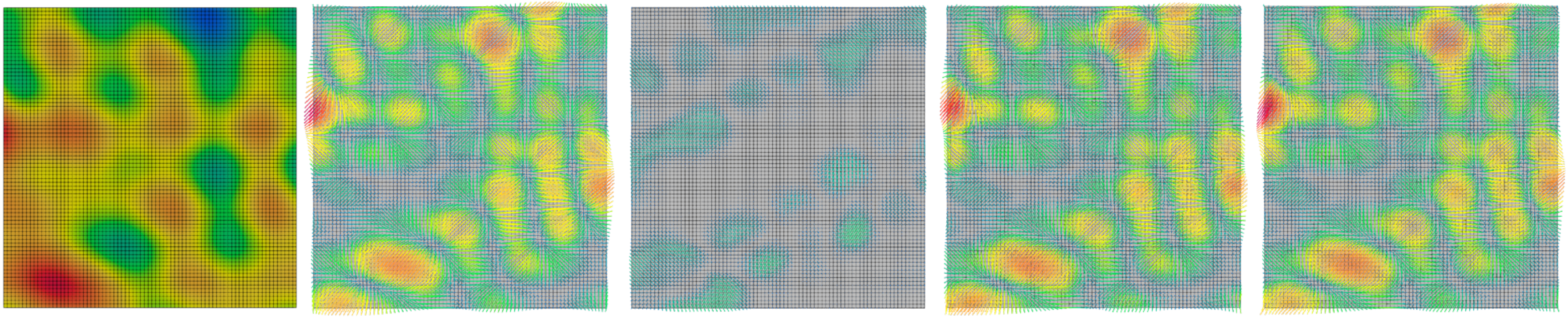
スカラー場
（入力）



勾配場
（正解）

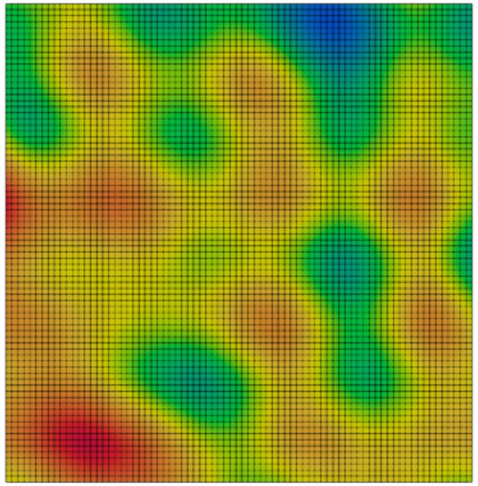
- X、Y 方向に 10 ～ 100 要素ある
  グリッドメッシュをランダムに生成
- ランダムなフーリエ級数で
  スカラー場を生成
- 微分を解析的に計算
- training、validation、test dataset
  はそれぞれ 100 サンプル

実験：微分演算子の学習

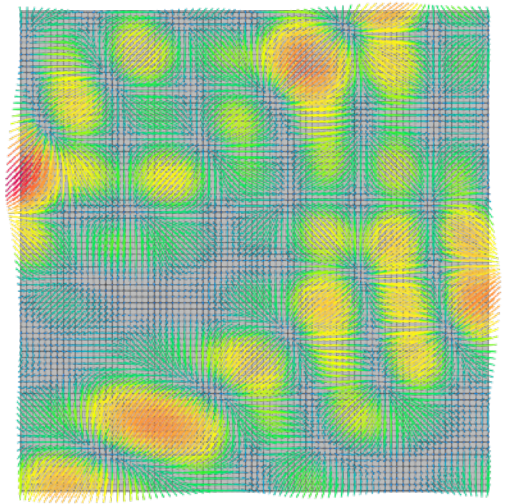


- 既存の GCN 系の手法では空間の情報を扱えないが IsoGCN では可能

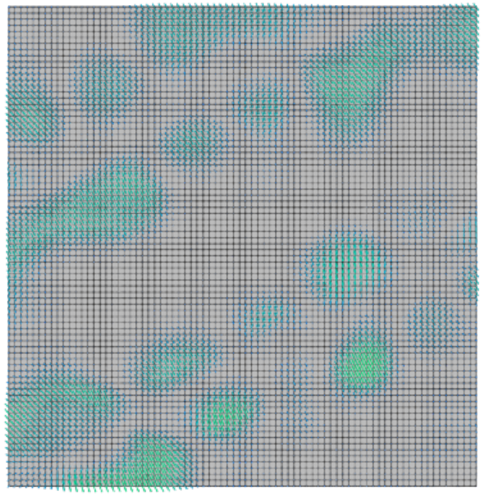
実験：微分演算子の学習



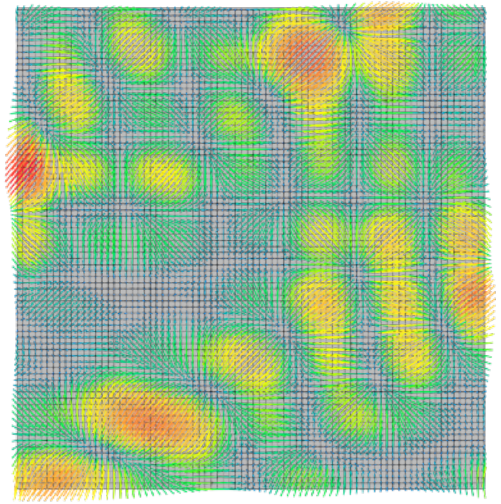
スカラー場
（入力）



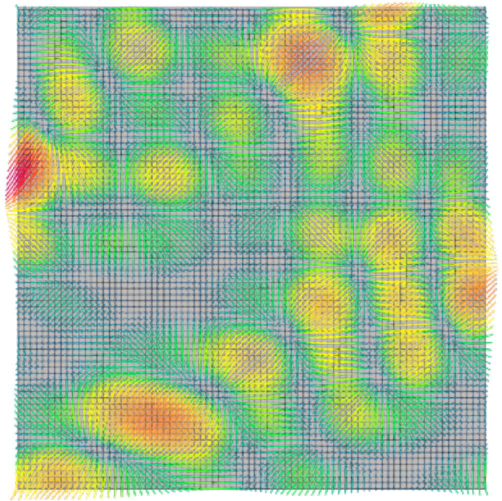
勾配場
（正解）



Cluster-GCN
（推論）



SE(3)-Trans.
（推論）



IsoGCN
（推論）

100 サンプルの推論時間：	4.0 s	0.4 s

- 既存の GCN 系の手法では空間の情報を扱えないが IsoGCN では可能

実験：微分演算子の学習

Method	# hops	$\boldsymbol{x}$	スカラー ↓ 勾配 **Loss of 0 → 1** $\times 10^{-5}$	スカラー ↓ ヘッシアン **Loss of 0 → 2** $\times 10^{-6}$	勾配 ↓ ラプラシアン **Loss of 1 → 0** $\times 10^{-6}$	勾配 ↓ ヘッシアン **Loss of 1 → 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	**1.79** ± 0.02	**3.50** ± 0.04	**2.52** ± 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	**1.44** ± 0.02
	5	No	14.19 ± 0.10	21.72 ± 0.25	34.09 ± 0.19	8.32 ± 0.09

- 既存の GCN 系の手法では空間の情報を扱えないが IsoGCN では可能

実験：微分演算子の学習

	スカラー ↓ 勾配 $\mathbf{0 \to 1}$		スカラー ↓ ヘッシアン $\mathbf{0 \to 2}$	
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

- IsoGCN は TFN、SE(3)-Transformer （TFN の改良）より高速

実験：非定常・異方的・非線型熱拡散問題

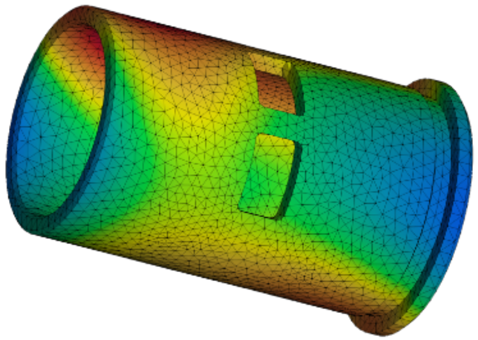
$$\Omega \subset \mathbb{R}^3,$$

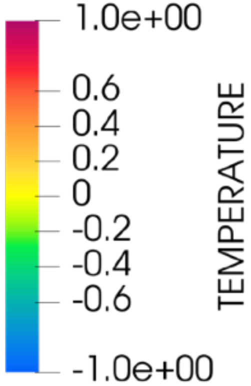
$$\frac{\partial T(\boldsymbol{x}, t)}{\partial t} = \nabla \cdot \boldsymbol{C}(T(\boldsymbol{x}, t)) \nabla T(\boldsymbol{x}, t) \quad \text{in } \Omega,$$

$$T(\boldsymbol{x}, t = 0) = T_{\text{init}}(\boldsymbol{x}) \quad \text{in } \Omega,$$

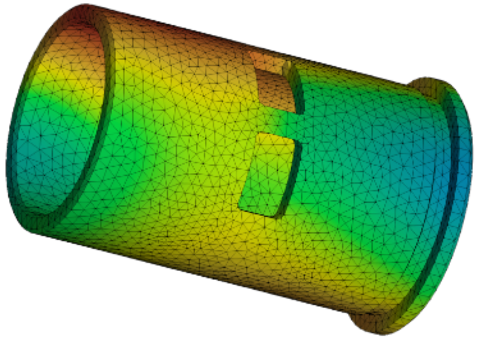
$$\nabla T(\boldsymbol{x}, t)\big|_{\boldsymbol{x}=\boldsymbol{x}_b} \cdot \boldsymbol{n}(\boldsymbol{x}_b) = 0 \quad \text{on } \partial\Omega$$

初期条件



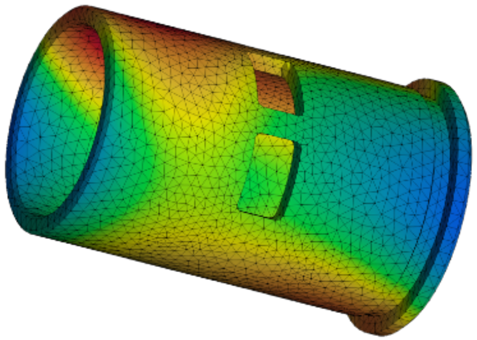


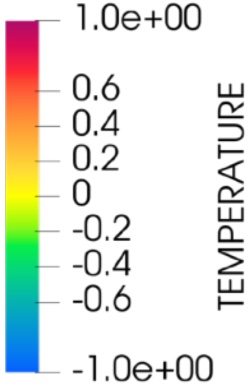
有限要素解析



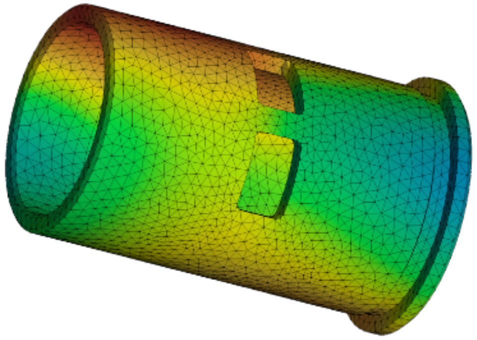
実験：非定常・異方的・非線型熱拡散問題

初期条件





有限要素解析



$$\Omega \subset \mathbb{R}^3,$$

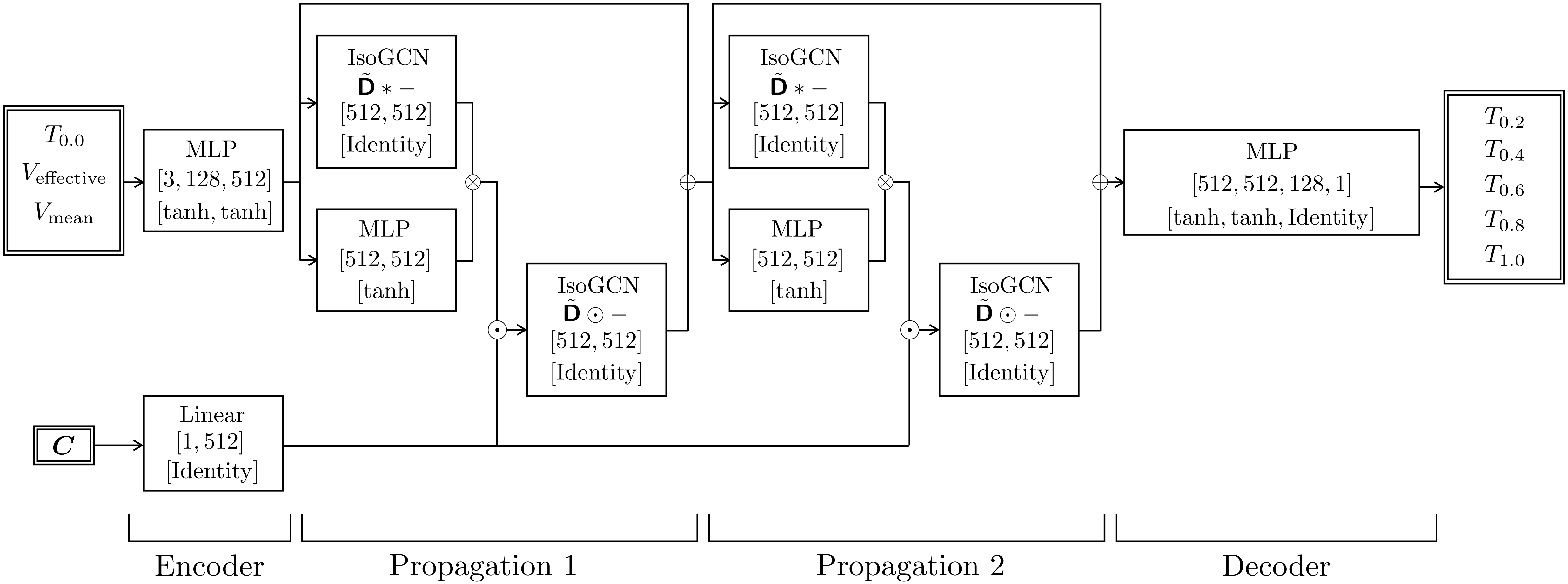
$$\frac{\partial T(\boldsymbol{x}, t)}{\partial t} = \nabla \cdot \boldsymbol{C}(T(\boldsymbol{x}, t)) \nabla T(\boldsymbol{x}, t) \quad \text{in } \Omega,$$

$$T(\boldsymbol{x}, t = 0) = T_{\text{init}}(\boldsymbol{x}) \quad \text{in } \Omega,$$

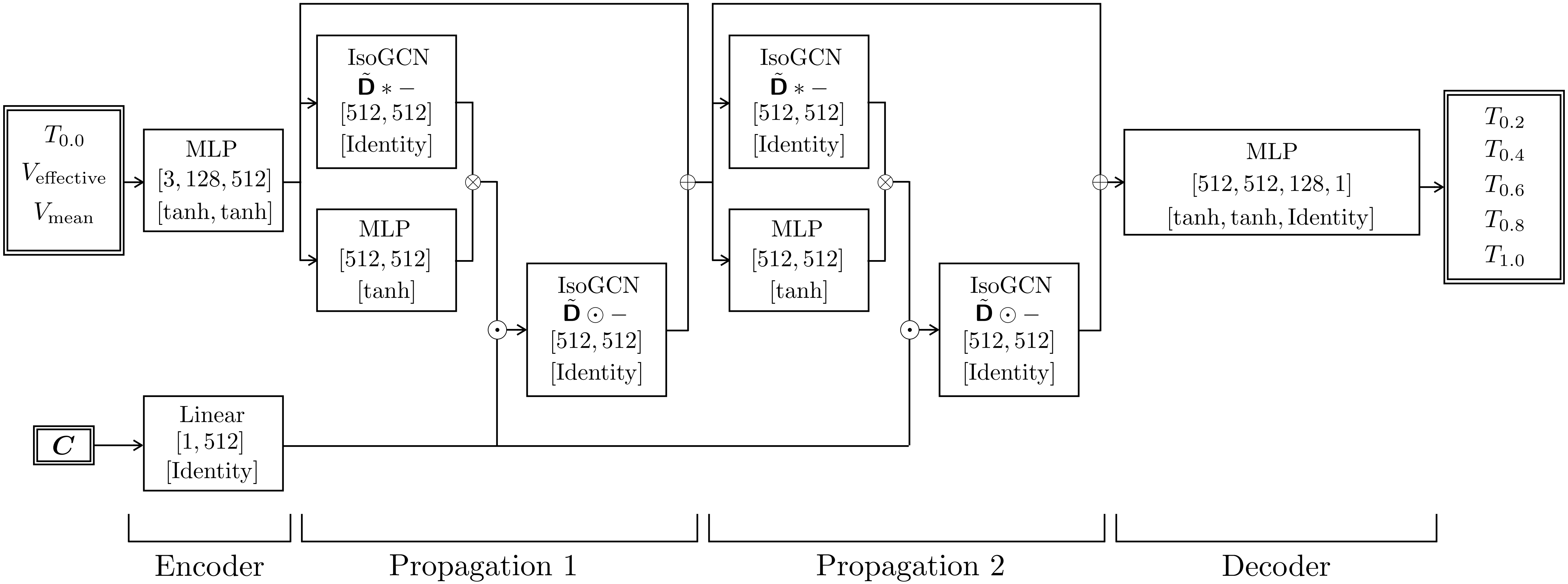
$$\nabla T(\boldsymbol{x}, t)|_{\boldsymbol{x}=\boldsymbol{x}_b} \cdot \boldsymbol{n}(\boldsymbol{x}_b) = 0 \quad \text{on } \partial\Omega$$

- ABC dataset (Koch+ 2018) から CAD データを抽出
- CAD 形状ひとつにつきメッシュサイズ 3 通り 初期条件 3 通りの最大 9 サンプルを生成
    - Training dataset: 50 形状、439 サンプル
    - Validation dataset: 16 形状、143 サンプル
    - Test dataset: 16 shapes、140 サンプル

実験：非定常・異方的・非線型熱拡散問題



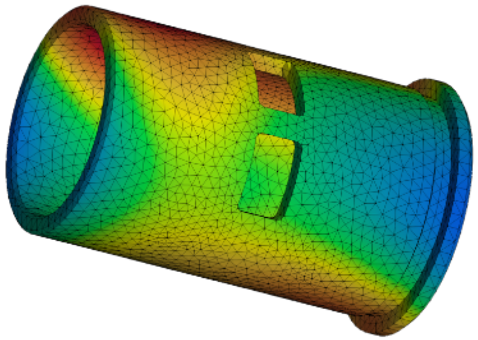
実験：非定常・異方的・非線型熱拡散問題

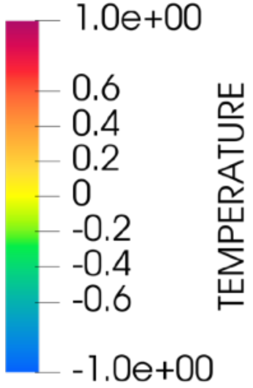


$$T(\boldsymbol{x}, t + \Delta t) \simeq T(\boldsymbol{x}, t) + \nabla \cdot \boldsymbol{C}(T(\boldsymbol{x}, t)) \nabla T(\boldsymbol{x}, t) \Delta t$$

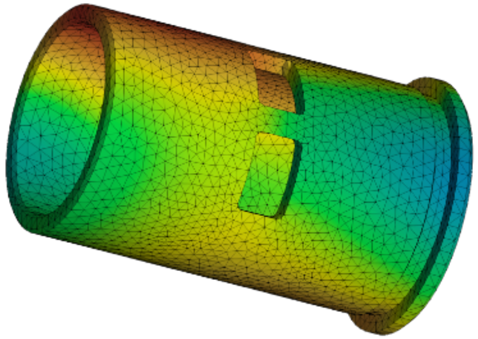
実験：非定常・異方的・非線型熱拡散問題

初期条件





有限要素解析

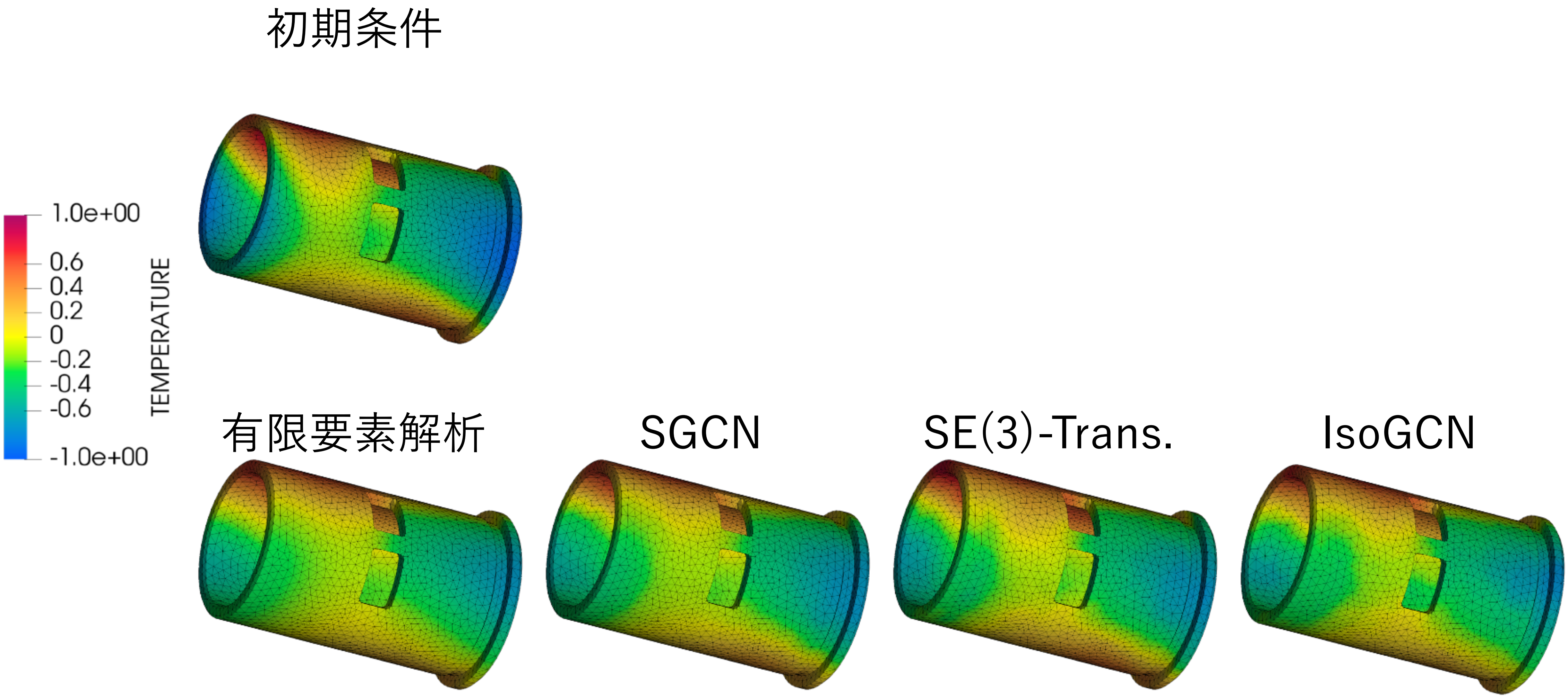


$$\Omega \subset \mathbb{R}^3,$$

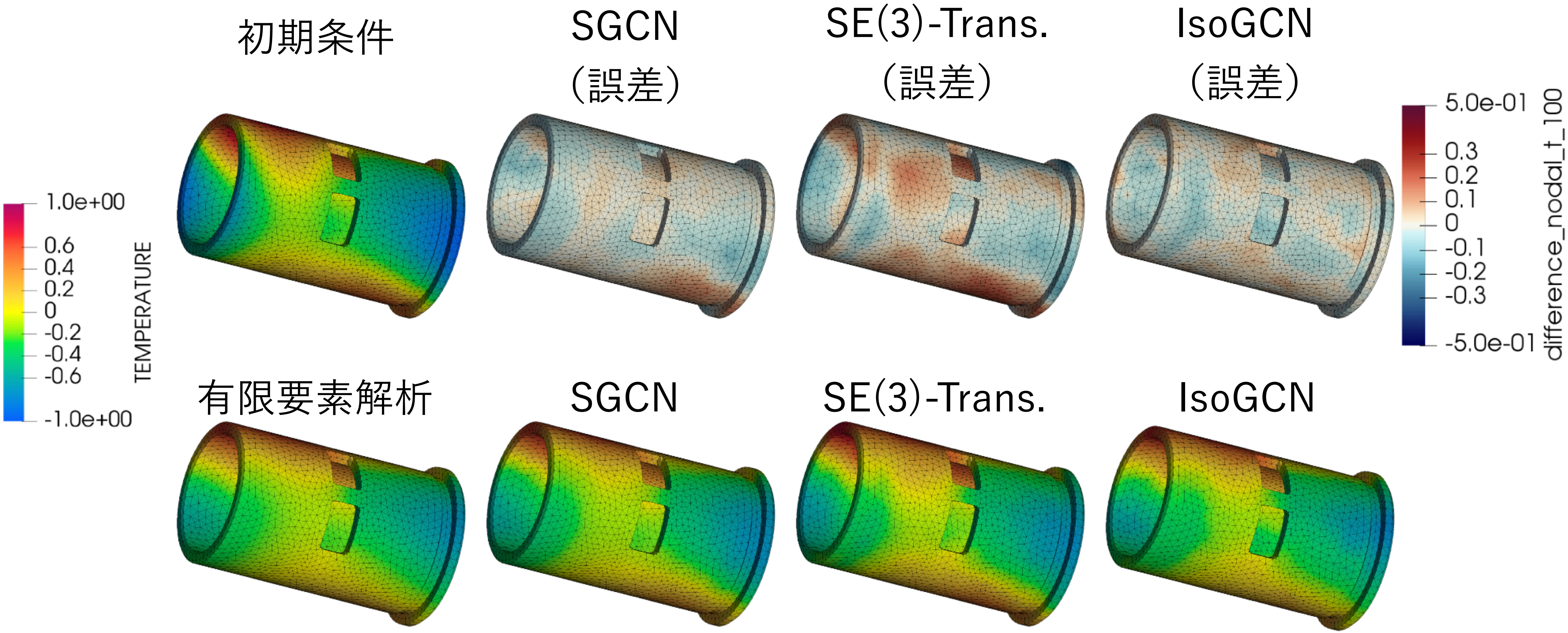
$$\begin{aligned}
\frac{\partial T(\boldsymbol{x}, t)}{\partial t} &= \nabla \cdot \boldsymbol{C}(T(\boldsymbol{x}, t)) \nabla T(\boldsymbol{x}, t) && \text{in } \Omega, \\
T(\boldsymbol{x}, t = 0) &= T_{\text{init}}(\boldsymbol{x}) && \text{in } \Omega, \\
\nabla T(\boldsymbol{x}, t)|_{\boldsymbol{x}=\boldsymbol{x}_b} \cdot \boldsymbol{n}(\boldsymbol{x}_b) &= 0 && \text{on } \partial\Omega
\end{aligned}$$

- ABC dataset (Koch+ 2018) から CAD データを抽出
- CAD 形状ひとつにつきメッシュサイズ 3 通り
  初期条件 3 通りの最大 9 サンプルを生成
  - Training dataset: 50 形状、439 サンプル
  - Validation dataset: 16 形状、143 サンプル
  - Test dataset: 16 shapes、140 サンプル

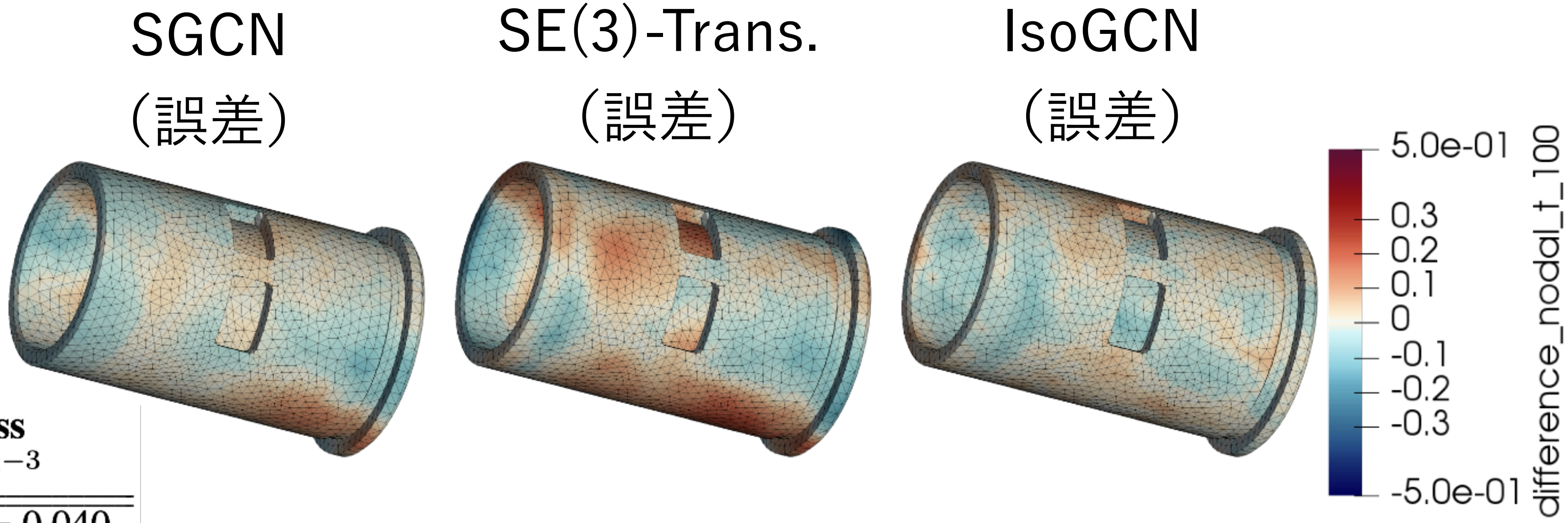
実験：非定常・異方的・非線型熱拡散問題



実験：非定常・異方的・非線型熱拡散問題



実験：非定常・異方的・非線型熱拡散問題



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	$\mathbf{2.470} \pm 0.008$

➡ IsoGCN は既存手法と比較して高精度

➡ ただし、TFN と SE(3)-Transformer はメモリに載せるためにパラメータ数を少なくしている

実験：非定常・異方的・非線型熱拡散問題

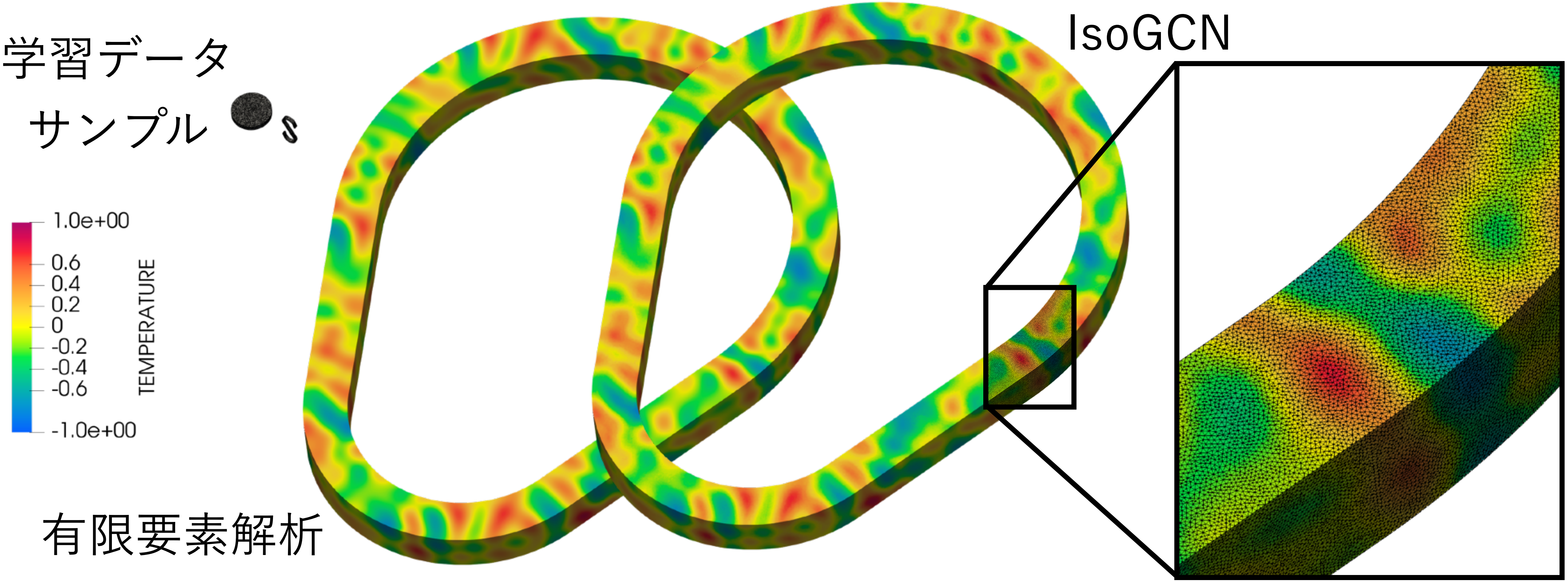
➡ IsoGCN は同等の精度となるセッティングの FrontISTR と比較して高速

➡ TFN・SE(3)-Transformer は大規模メッシュでは Out-of-Memory (500 GB)

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**

実験：非定常・異方的・非線型熱拡散問題

- 学習データサンプルより大きなメッシュに対しても外挿可能



目次

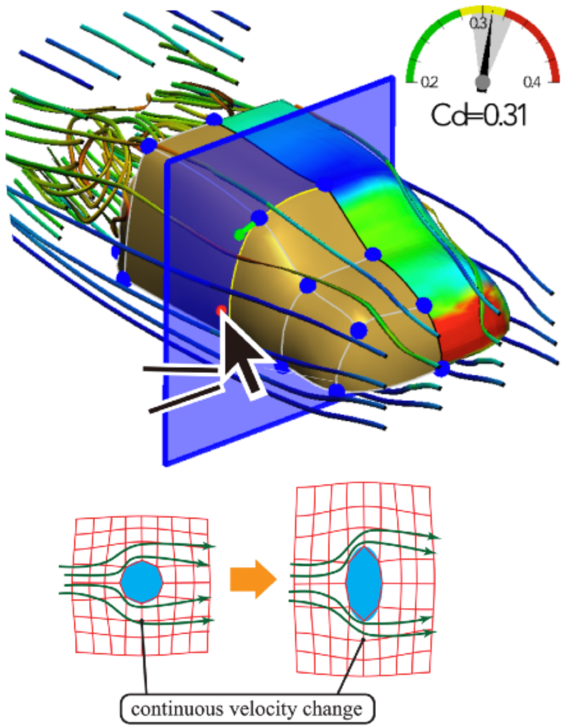
- Graph neural network
- 同変性
- IsoGCN: 同変性＋Graph Neural Network
- まとめ

目次

- Graph neural network
- 同変性
- IsoGCN: 同変性＋Graph Neural Network
- まとめ

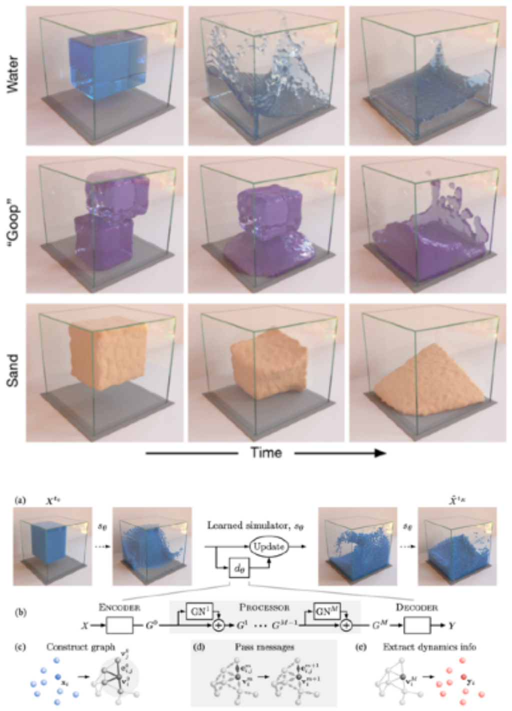
関連研究：物理シミュレーションの機械学習

→ Graph Neural Network の適用　→ 同変性の導入



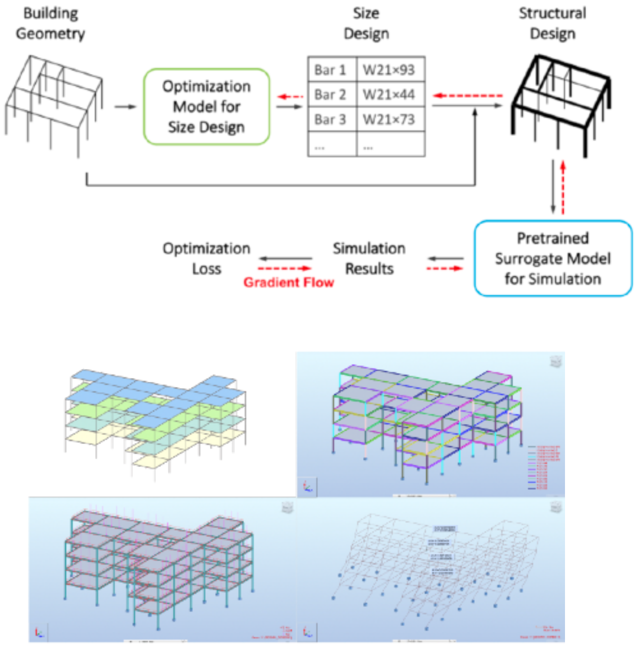
Umetani+ 2018

- リアルタイムで
  車体の空力性能予測
- メッシュ構造固定



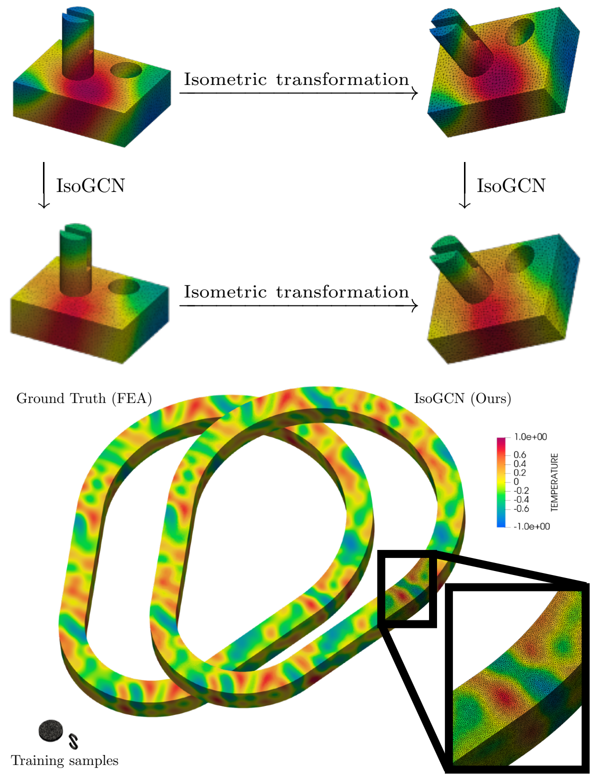
Sanchez-Gonzalez+ 2020

- 任意点群での学習
- 高速化は未達成



Chang+ 2020

- 任意梁構造での学習
- 梁部材選定の最適化

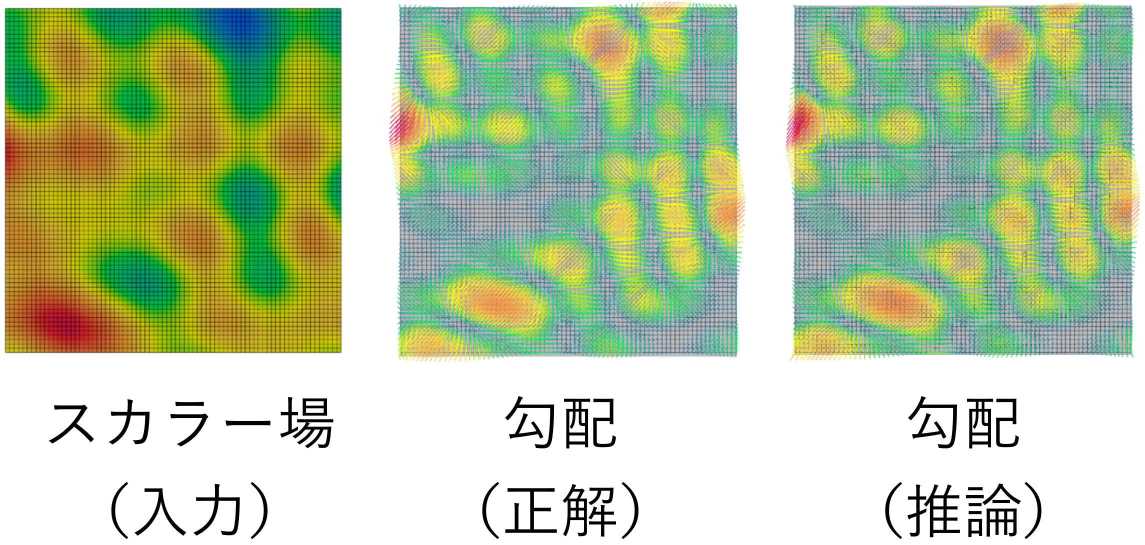


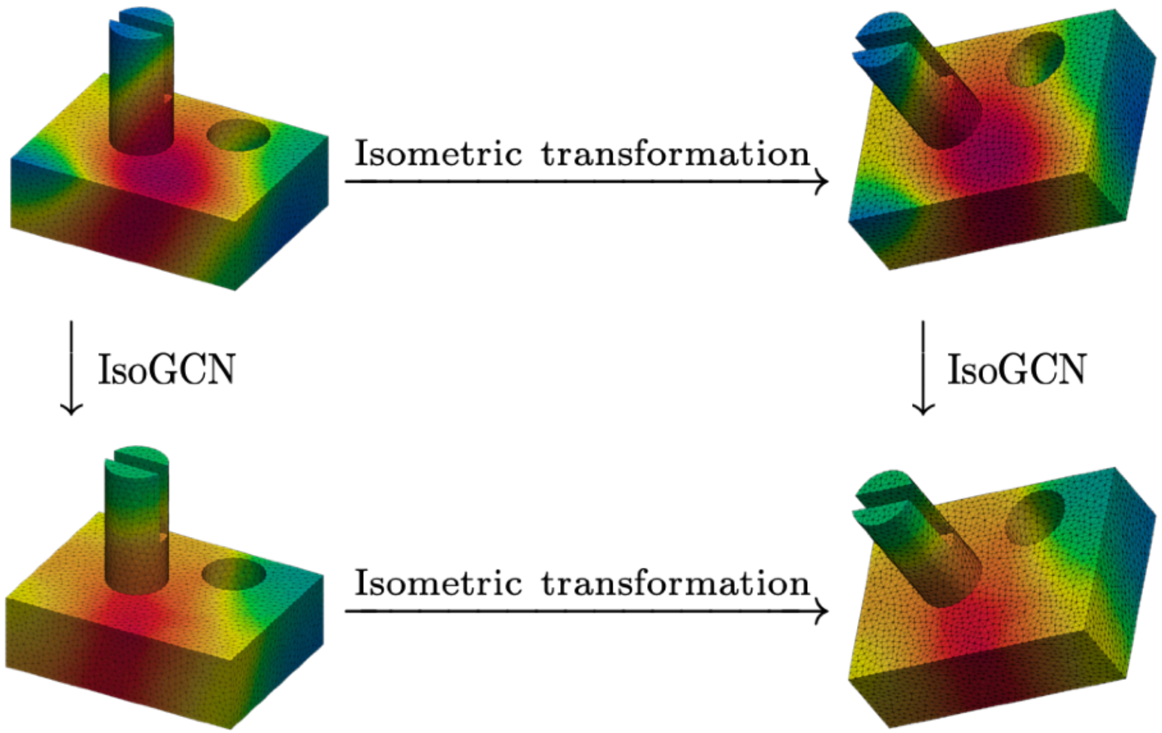
Horie+ 2020

- 任意メッシュでの学習
- 高速化を達成
- 物理現象の対称性を考慮

概要：対称性があり計算効率のよい GNN を開発

- IsoGCN: 合同変換不変・同変で計算効率のよい GNN (Horie+ 2020)
- IsoGCN は同等の精度の有限要素解析より高速
- IsoGCN は微分演算子の近似とみなせる



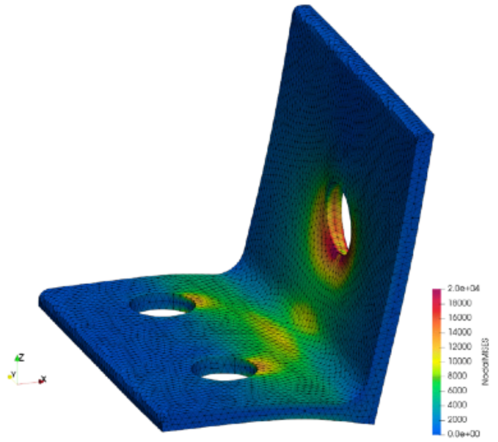
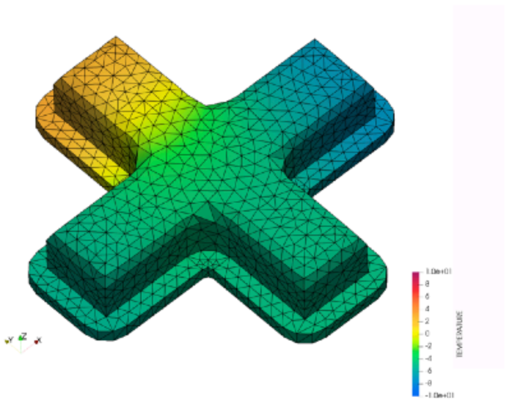


概要：対称性があり計算効率のよい GNN を開発

- IsoGCN: 合同変換不変・同変で計算効率のよい GNN (Horie+ 2020)
- IsoGCN は同等の精度の有限要素解析より高速
- IsoGCN は微分演算子の近似とみなせる

今後の展望：

- IsoGCN の近似性能を評価
- IsoGCN を多様な物理現象へ適用


References

- N. Umetani, and B. Bickel. Learning three-dimensional flow for interactive aerodynamic design. ACM Transactions on Graphics (TOG) 37.4: 1-10, 2018.
- A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. W. Battaglia. Learning to simulate complex physics with graph networks. In ICML, 2020.
- K. Chang and C. Cheng. Learning to simulate and design for structural engineering. In ICML, 2020.
- M. Horie, N. Morita, T. Hishinuma, Y. Ihara, and N. Mitsume. Isometric Transformation Invariant and Equivariant Graph Convolutional Networks. arXiv preprint arXiv:2005.06316, 2020.
- E. Ahmed, A. Saint, A. E. R. Shabayek, K. Cherenkova, R. Das, G. Gusev, D. Aouada and B. Ottersten. Deep Learning Advances on Different 3D Data Representations: A Survey. arXiv preprint arXiv:1808.01462, 2018.
- T. N. Kipf, and M. Welling. Semi-supervised classification with graph convolutional networks. In ICLR, 2017.
- N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. arXiv preprint arXiv:1802.08219, 2018.
- N. Dym, and H. Maron. On the Universality of Rotation Equivariant Point Cloud Networks. arXiv preprint arXiv:2010.02449, 2020.
- T. Tamai and S. Koshizuka. Least squares moving particle semi-implicit method. Computational Particle Mechanics, 1(3):277–305, 2014.
- S. Koch, A. Matveev, Z. Jiang, F. Williams, A. Artemov, E. Burnaev, M. Alexa, D. Zorin, and D. Panozzo. Abc: A big cad model dataset for geometric deep learning. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.