

境界面の陰関数表現を導入した Graph Neural Network の構築

Development of a Graph Neural Network
Incorporating Implicit Function Representations of Boundaries

権太祐樹¹⁾ 三目直登²⁾
Yuki Gonda, Naoto Mitsume

¹⁾筑波大学大学院システム情報工学研究群 (〒 305-8573 茨城県つくば市天王台 1-1-1)

²⁾筑波大学システム情報系 (〒 305-8573 茨城県つくば市天王台 1-1-1)

In this study, we conducted a foundational investigation into the introduction of machine learning models as an alternative to numerical analysis, aimed at addressing the discrepancies observed between analytical solutions and real-world phenomena. Specifically, we developed and adapted two models, PointwiseMLP and Graph Neural Network (GNN), for temperature field predictions, comparing their predictive accuracy. These models were trained using solutions obtained from the Smoothed Particle Hydrodynamics (SPH) method as training data. We evaluated which model achieved higher predictive accuracy across various scenarios, including changes in the initial heat source location. The results demonstrated that the GNN model consistently surpassed the PointwiseMLP model in predictive accuracy across all tested datasets, achieving an average L2 error reduction rate of approximately 64.4%. This suggests that GNN's capability to incorporate surrounding information significantly enhances its predictive accuracy. Additionally, deterioration in predictive accuracy near the boundary areas was observed, leading us to introduce a boundary's implicit function representation using particle number density, into both models to improve predictive accuracy near boundaries and aim for an expansion of generality towards geometric shapes.

Key Words : Surrogate modelling, Graph Neural Network, Implicit Function Representations

1. 序論

有限要素法をはじめとする数値解析手法は産業界などの幅広い領域で重要な役割を担っている。特に、製品開発過程における製品の評価・最適化のために広く利用されている。しかし、数値解析の過程で「数値解と実現象との乖離が見られる」、「対象によっては解析に膨大な時間がかかる」といった問題が起きている。これらの問題を解決する技術として近年、機械学習が注目されており、その数値解析代替に関する研究は 1990 年代 [1-4] から行われている。物理現象の予測に機械学習を用いる利点として、現象を表す背後の方程式やその離散化手法等の情報が未知、あるいは想定外に高精度な方程式や離散化手法が必要である場合において、実験・実測データを用いた学習により、学習器がそのような情報を獲得可能であることが挙げられる。機械学習による予測は、実験・実測データを用いた学習を行うという条件下で、複雑現象、あるいは未知現象における既存の数値解析による解よりも高精度な予測解を算出可能であると期待される。また、機械学習モデルの事前学習により、数値解析と比較した計算時間の短縮も見込まれる。実際に、物理現象の性質を内部に組み込み、大域的な相互作用と境界条件を考慮可能な機械学習モデルである Physics-Embedded Neural Networks により流体現象の予測を行った先行研究 [5] では、高いレベルでの計算速度・予測精度のトレードオフが実現できることが示されている。このような、数値解析代替となる高精度かつ汎用的な機械学習モ

ルを構築することが本研究のモチベーションとなっている。

物理現象を対象とした予測を行う場合、すべての対象において、自由度や幾何形状が同一であることは考えにくい。そのため、予測に使用する機械学習モデルは、自由度・幾何形状に対する汎用性を持つことが必要である。本研究では、任意自由度・形状に適応可能な機械学習モデルである、Pointwise Multi-Layer Perceptron (PointwiseMLP) と PointwiseMLP に message 関数を導入した Graph Neural Network(GNN) [6] [7] の温度場解析における予測精度の比較を行う。対象は偏微分方程式であり、微分は周囲との値の差に相当するため、本来 1 点の情報のみから解を算出できない。周囲の情報を考慮して予測を行う GNN の方が、1 点のみの情報から予測を行う PointwiseMLP に対して予測精度の面で優れていると予想される。実際に、数値解析による解析結果を GNN で学習し、数値解析の代替を試みる研究は近年盛んに行われており [8] [9]、粒子法の代替モデルとして流体現象の予測に適応させた先行研究 [9] では、GNN が解析対象の自由度・形状に対する汎用性を持つだけでなく、材質や粘度の異なる流体同士が相互作用する現象も予測可能なことが示されている。本研究の目的を、従来の数値解析の代替となる高精度かつ汎用的な機械学習モデルを開発するための基礎的検討とする。そのようなモデルとして、任意自由度・形状に対する汎用性を持つ点で適切だと考えられる、PointwiseMLP, GNN において

同条件での予測精度を比較することで本研究の目的を達成する。また、両モデルに境界の陰関数表現である界面関数を入力することで、境界付近の予測精度を向上させ、幾何形状に対する汎用性の拡張を目指す。

2. 物理現象の予測に使用する機械学習モデル

(1) 点群データのグラフ構造化

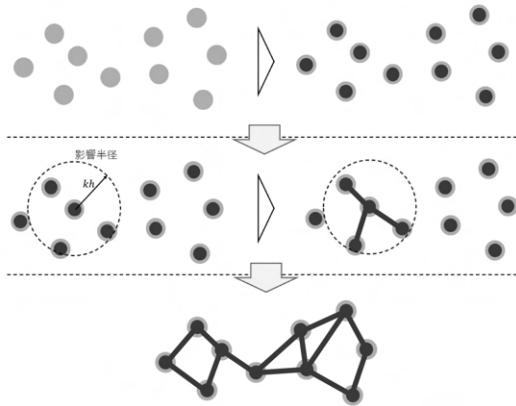


図-1 点群データのグラフ構造化

本研究では粒子法である Smoothed Particle Hydrodynamics (SPH) 法による数値解を教師データとして用いるため、機械学習モデルは点群データを扱う。点群データは図 1 によってグラフ構造に変換することが可能である。まず、生成した各粒子をグラフ理論の頂点（ノード）として捉える。次に、各粒子の計算には影響半径内に存在する粒子の情報を用いるため、各粒子の影響半径内に存在する粒子と辺（エッジ）で結ぶ。すべての粒子に同様の操作を行うことで、グラフを生成できる。本研究では、このような点群データのグラフ構造を対象として機械学習モデルを用いた学習と予測を考える。

(2) 粒子数密度を用いた境界の陰関数表現

粒子法における解析空間において、境界の情報を陰に表現するために、粒子数密度を用いた境界の陰関数表現を考える。

$$\phi_i = \frac{n_i}{n_0} \quad (1)$$

$$n_i = \sum_j W(x_i - x_j, h) \quad (2)$$

粒子 i における粒子数密度を n_i 、粒子位置を x_i とする。SPH 法における重み関数を W 、影響半径を h とする。粒子数密度 n_i はある粒子 i の周囲において、どの程度の密度で粒子が分布しているかを表すための指標である。境界付近でなく、十分内部に位置する粒子における粒子数密度を基準値 n_0 とする。ここで、 ϕ_i は $0 < \phi_i < 1$ の範囲の実数値をとる粒子数密度を用いた境界の陰関数表現であると言える。

(3) PointwiseMLP

PointwiseMLP は、各点に同一の Multi-Layer Perceptron (MLP) を用いるため、任意形状・自由度に対応可能

である。中間層と出力層を合わせた層数が L 層の MLP を $MLP_L : \mathbb{R}^{d_{in}} \rightarrow \mathbb{R}^{d_{out}}$ とする。PointwiseMLP _{L} : $\mathbb{R}^{|V| \times d_{in}} \rightarrow \mathbb{R}^{|V| \times d_{out}}$ は、同一の MLP_L を各点に適用することによって構築される。 V は点群をグラフ構造とみなしたときの頂点集合、 $|V|$ は全頂点数を表す。なお、本項と (4) 項を執筆するにあたり、堀江氏による博士論文 [10] の定式化を参考にさせていただいた。

$$\text{PointwiseMLP}_L(\mathbf{H}_{in}) := \begin{pmatrix} MLP_L(h_{in,1}) \\ MLP_L(h_{in,2}) \\ \vdots \\ MLP_L(h_{in,|V|}) \end{pmatrix} \quad (3)$$

$$\mathbf{H}_{in} = \begin{pmatrix} h_{in,1} \\ h_{in,2} \\ \vdots \\ h_{in,|V|} \end{pmatrix} \in \mathbb{R}^{|V| \times d_{in}} \quad (4)$$

$$h_{in,i} = \begin{pmatrix} u_i \\ \phi_i \end{pmatrix} \in \mathbb{R}^2 \quad (5)$$

$h_{in,i}, \forall i \in \{0, 1, \dots, |V|\}$ は頂点 i の入力特微量ベクトルを、 $h_{out,i}$ は頂点 i の出力特微量ベクトルを表す。ここで、 u_i は頂点 i における温度を、 ϕ_i は式 (1) に示した、粒子数密度を用いた境界の陰関数表現を表す。各点に作用させる MLP_L は以下のように表される。

$$\begin{aligned} h_{out,i} &= MLP_L(h_{in,i}) \\ &:= \sigma^{(L)} \circ \text{Affine}^{(L)} \\ &\quad \circ \dots \circ \sigma^{(1)} \circ \text{Affine}^{(1)}(h_{in,i}) \end{aligned} \quad (6)$$

PointwiseMLP では同一の MLP_L を各頂点に対して使い回すため、図 2 のように頂点数が変化しても予測可能である。このため、PointwiseMLP は任意形状・自由度に対して汎用性があると言える。しかし、グラフ同士の接続情報（エッジ）を無視しているため、更新点の周囲の情報を考慮できないという問題点がある。この問題を解決した機械学習モデルが Graph Neural Network(GNN) である。

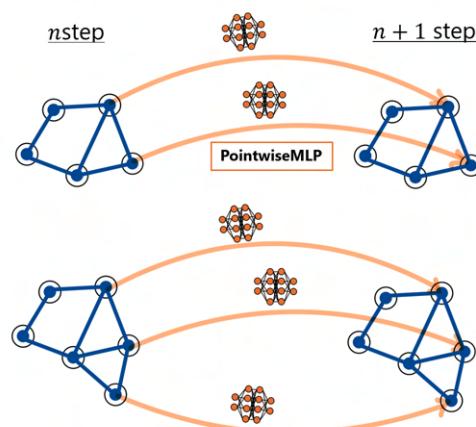


図-2 異なる幾何形状の取り扱い: PointwiseMLP を用いた予測の場合

(4) GNN

GNN は特にグラフ構造に対して用いるニューラルネットワークの総称である。様々な種類の GNN が提案されているが、本研究では Message Passing Neural Network(MPNN) [7] の枠組みを GNN として捉える。GNN では式(7), 式(8)のように、Message 関数 f_{message} による周囲の情報の集約と Update 関数 f_{update} による情報の更新という 2 段階の処理に分かれている。

$$\mathbf{h}_{\text{out},i} := f_{\text{update}}(\mathbf{h}_{\text{in},i}, \mathbf{m}_i) \quad (7)$$

$$\mathbf{m}_i := \sum_{j \in N_i} f_{\text{message}}(\mathbf{h}_{\text{in},i}, \mathbf{h}_{\text{in},j}, e_{\text{in},ij}), \quad (8)$$

ここで、 $\{\mathbf{h}_{\text{out},i}\}_{i \in V}$ と $\{e_{\text{in},ij}\}_{(i,j) \in E}$ はそれぞれ頂点特徴量とエッジ特徴量である。なお、 E は点群をつなぐエッジの集合を、 N_i は頂点 i とエッジで接続されている頂点集合を表す。エッジ特徴量 $\{e_{n_i,j}\}_{(i,j)}$ は、エッジ上に付与される情報である。 f_{update} は PointwiseMLP と同様の構造をとり、 f_{message} は多層ニューラルネットワーク構造となる。なお、 $f_{\text{update}}, f_{\text{message}}$ の入力特徴量 $\mathbf{h}_{\text{in},i}$ は式(5) と同様の構造とする。

図3にGNNによる予測方法を示している。Message関数とUpdate関数はそれぞれ同一のMLPを使い回しているため、GNNが処理可能なグラフ構造は、頂点数またはエッジの本数に依存しない。よって、GNNもPointwiseMLPと同様に自由度・形状に対する汎用性を持つ。

PointwiseMLPは周囲の情報を考慮できないという欠点を持っていたが、GNNはMessage関数により周囲の情報を考慮しているため、より高精度な予測が可能であると考えられる。

式(5)に示したように入力特徴量に境界の陰関数表現を入力することで、PointwiseMLP, GNNは予測対象とする点が境界に位置するかどうかの情報を獲得することが期待される。

3. 問題設定と結果・検証

(1) 問題設定

本項では、熱拡散問題における温度予測へのPointwiseMLPとGNNの適用方法について述べる。

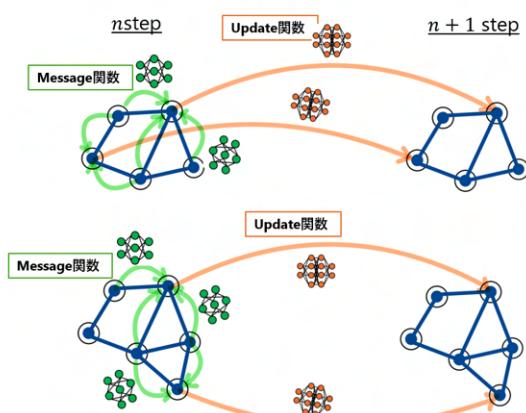


図3 異なる幾何形状の取り扱い: GNN を用いた予測の場合

a) 教師データ

教師データとして、開発した SPH 法熱拡散方程式ソルバーによる数値解を用いる。よって対象の偏微分方程式は、式(9)の熱拡散方程式である。

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u \quad (9)$$

u を温度 K, α を熱拡散係数 m^2/s とする。教師データの解析領域、Dirichlet 境界条件は以下のように設定している。

- 解析領域: $0 \leq x \leq 1, 0 \leq y \leq 1$
- Dirichlet 境界条件: $u(x, y) = 0$ at $(x = 0, 1 \text{ or } y = 0, 1)$

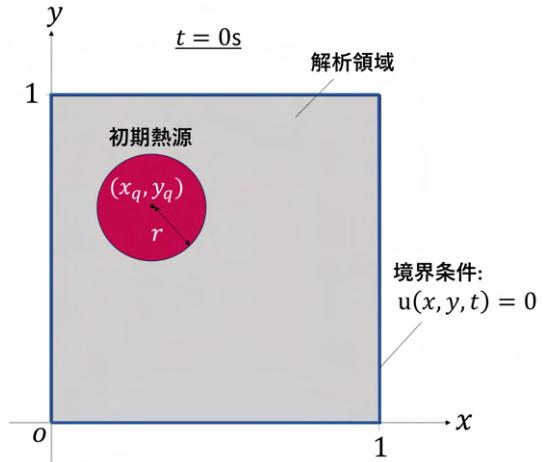


図4 初期熱源のプロット

初期条件として、図4に示したように、解析領域内に半径 r の熱源が $t = 0\text{s}$ のときのみ瞬間に現れる条件を定義する。本研究ではこのような熱源を初期熱源と呼ぶ。初期熱源のパラメータは表1で定義される。なお、式(10), 式(11), 式(12)に基づき、変数 r, x_q, y_q は定義される。

表1 初期熱源のパラメータ設定

Parameter	Value	Unit
温度	$0 \leq u(x, y) < 100$	K
熱源半径	$\frac{3}{41} \leq r < \frac{5}{41}$	m
中心点の x 座標	$r \leq x_q < (1 - r)$	
中心点の y 座標	$r \leq y_q < (1 - r)$	

$$r = \frac{1}{41} \left(3 + \frac{2\gamma}{32768} \right) \quad (10)$$

$$x_p = r + \frac{(1-r)\eta}{32768} \quad (11)$$

$$y_p = r + \frac{(1-r)\kappa}{32768} \quad (12)$$

$\beta, \gamma, \eta, \kappa$ はそれぞれ 0 から 32767 の範囲でそれぞれ異なるランダムな実数値として定められる。乱数の種として、プログラム実行時の UNIX 時刻 (1970 年から現在までの経過秒数) を用いている。なお、教師データ作成時

は毎セットごとに異なる時刻でプログラムを実行しているため、それぞれのデータセット作成時の乱数の種は異なる。

0.1sにおける温度分布を機械学習への入力データ、0.2sにおける温度分布を正解データとしてデータセットの作成を行った。教師データセットを作成する上で設定したパラメータを以下の表2に記載した。

表-2 教師データセットのパラメータ設定

Parameter	Value	Unit
粒子数	41×41	
時間刻み幅	0.1	s
熱拡散係数	0.0112	m^2/s
データセット数	1200	

b) 機械学習モデル内部の詳細設定

表3、表4にPointwiseMLP、GNNのハイパーパラメータの設定を記述している。また、PointwiseMLPとGNN内部で設定した活性化関数、損失関数、最適化手法の種類について表5に明記した。

表-3 PointwiseMLP のハイパーパラメータ設定

Parameter	Value
入出力特徴量次元	1
中間層の層数	6
中間層の次元	8

表-4 GNN のハイパーパラメータ設定

Parameter	Value
入出力特徴量次元	1
Message 関数の中間層の層数	3
Message 関数の中間層の次元	8
Update 関数の中間層の層数	3
Update 関数の中間層の次元	8

表-5 PointwiseMLP と GNN 内部の共通設定

Parameter	Value
中間層の活性化関数	ReLU 関数
出力層の活性化関数	恒等関数
損失関数	MSELoss
最適化手法	Adam

c) PointwiseMLP, GNN による温度場予測方法

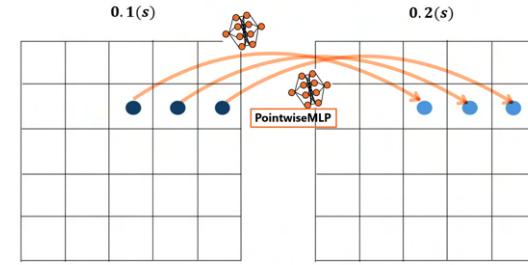


図-7 温度場予測への適応方法: PointwiseMLP を用いた場合

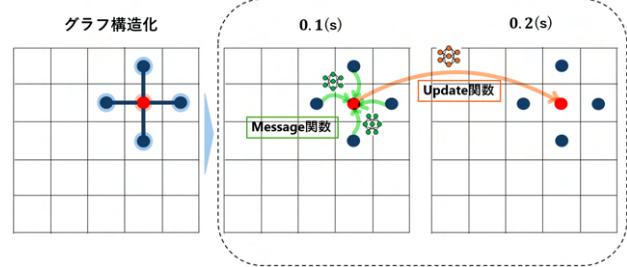


図-8 温度場予測への適応方法: GNN を用いた場合

PointwiseMLP と GNN のモデル実装には、オープンソースの機械学習ライブラリである PyTorch [11] を使用した。

まず、PointwiseMLP を用いた予測方法を解説する。(3) 項で言及したように、PointwiseMLP は各点ごとに同一の MLP を用いて予測を行う。入力特徴量として、式(5)に示したような、0.1sにおける各点の温度と境界の陰関数表現からなる 2 次元ベクトル $h_{in,i}$ を、出力特徴量として 0.2sにおける各点の温度 $h_{out,i}$ を出力する。図7に示すように、PointwiseMLP では各頂点の値が個別に更新される。

一方、GNN では図8に示したように、以下の3つのステップで予測を実行する。まず、更新対象の点を中心としたグラフ構造を構築する。次に、構築したグラフ構造に対し、Message 関数により 0.1秒における周囲の頂点情報を更新点に集約する。最後に、Update 関数により更新点の温度を 0.2秒の値に更新する。今回はグラフ構造においてエッジの情報を付与していないため、以下の式(14)に示すように、Message 関数にエッジの情報は入力しない。式(13)および式(14)において、 $h_{in,i}$ 、 $h_{in,j}$ は、それぞれ 0.1sにおける更新点と周囲の頂点での温度と境界の陰関数表現の 2 次元ベクトルを表す。 $h_{out,i}$ は 0.2sにおける更新点の温度を示す。

$$h_{out,i} := f_{update}(h_{in,i}, m_i) \quad (13)$$

$$m_i := \sum_{j \in N_i} f_{message}(h_{in,i}, h_{in,j}), \quad (14)$$

本稿では、紙面の都合上、入力特徴量を温度に限定した結果のみを提示する。PointwiseMLP および GNN に式(1)に示される境界の陰関数表現を導入した際の結果については、講演にて共有する。

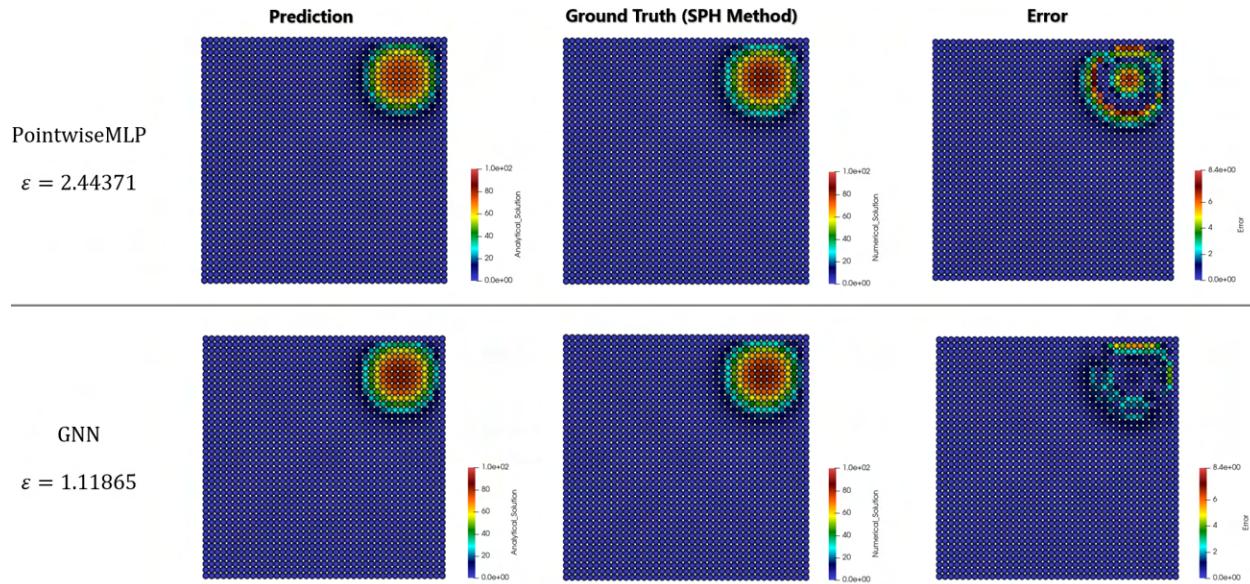


図-5 検証結果: 境界付近に初期熱源がある場合の PointwiseMLP と GNN の予測精度比較

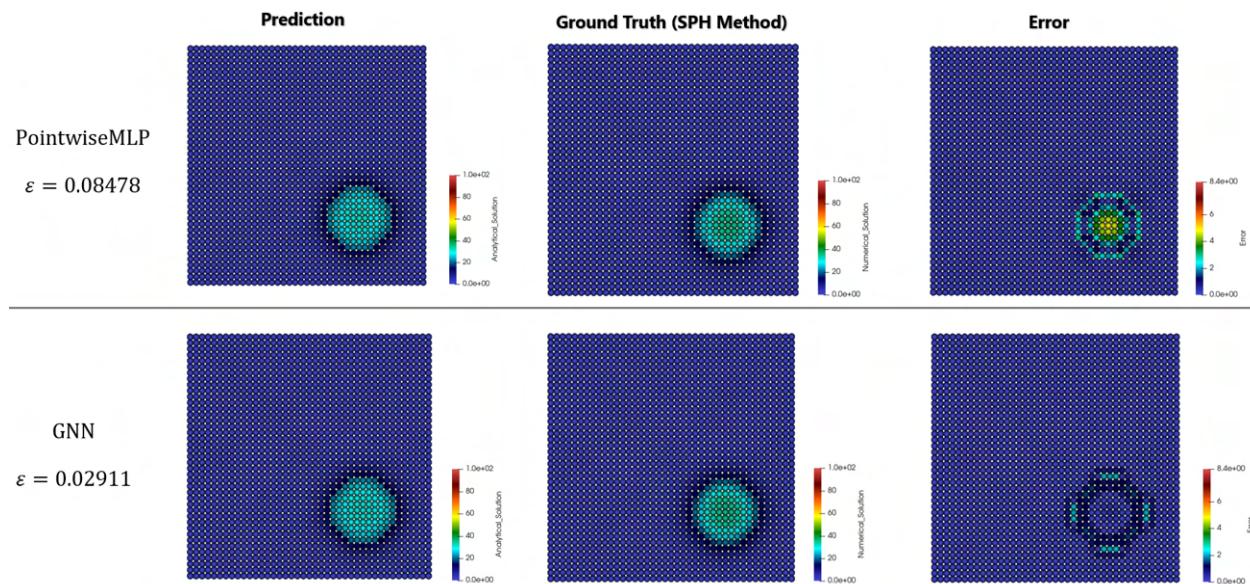


図-6 検証結果: 内部に初期熱源がある場合の PointwiseMLP と GNN の予測精度比較

(2) 結果

PointwiseMLP と GNN の温度場予測における両者の予測精度比較結果を示す。比較検証のために、初期熱源の温度、半径、位置が異なる 200 セットの検証データを作成した。2-a) 項で示した教師データを用いて、PointwiseMLP と GNN をそれぞれ学習させた後、検証データセットの 0.1s における温度分布を入力し、0.2s での温度分布を予測した。初期熱源が境界付近にある場合の予測結果を図 5 に、境界から離れている場合の結果を図 6 に示す。また、検証データ 200 セットの L2 誤差平均値を表 6 に、各セットにおける GNN の L2 誤差 ε_{GNN} から、PointwiseMLP の L2 誤差 $\varepsilon_{\text{PointwiseMLP}}$ を引いた値を可視化したグラフを図 9 に示す。図 9 では、縦軸の値が 0 未

満の場合、GNN の方が予測精度が優れていることを示す。ここで、L2 誤差 ε は式 (15) により計算した。 h_i^{true} は SPH による数値解、 h_i^{output} は PointwiseMLP 及び GNN による予測解、 N は解析領域内の全頂点数である。

$$\varepsilon = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N |h_i^{\text{true}} - h_i^{\text{output}}|^2}}{\sqrt{\frac{1}{N} \sum_{i=1}^N |h_i^{\text{true}}|^2}} \quad (15)$$

図 5、図 6 における誤差分布は、以下の式 (16) に示す、絶対値誤差により計算している。 h^{true} を SPH 法の数値解、 h^{output} を予測解とすると、粒子 i における絶対値誤差 ε_i は以下の式 (16) のように表される。

$$\varepsilon_i = |h^{\text{output}} - h^{\text{true}}| \quad (16)$$

表-6 検証結果: PointwiseMLP と GNN の全検証データにおける L2 誤差平均値

Property	Value
PointwiseMLP の L2 誤差	0.280729
GNN の L2 誤差	0.100003

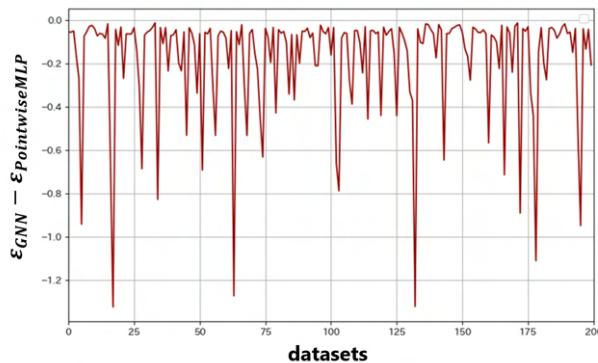


図-9 検証結果: 検証データ 200 セットそれぞれにおける $\varepsilon_{\text{GNN}} - \varepsilon_{\text{PointwiseMLP}}$ のプロット

(3) 考察

図 5 および図 6 に示した PointwiseMLP と GNN の予測解に対する誤差分布を比較すると、境界付近と内部領域のいずれにおいても GNN の方が高い精度を達成していることがわかる。この傾向は、図 9 に示した全検証データセットにおける比較からも裏付けられる。ここで縦軸は、全ての場合で負の値となっていることから、GNN の予測精度が一貫して上回っていると言える。さらに表 6 を見ると、GNN の L2 誤差平均値は PointwiseMLP と比べて約 64.4% 減少していることがわかる。

以上より、本研究における解析条件下では PointwiseMLP と比較して、GNN の方が予測精度が高いことが示された。重みパラメータの総数が PointwiseMLP, GNN で同程度にも関わらず GNN が PointwiseMLP よりも高精度な予測を達成できた要因は、GNN が Message 関数により周囲の情報を考慮できているためであると考えられる。ただし、図 5 の GNN の誤差分布からは、境界付近で精度が悪化する傾向も確認された。

この問題への対策として、本研究では境界の陰関数表現をモデルの入力に加えることを提案する。境界情報を陽に与えることで、境界付近の予測精度の改善が期待できる。そこで、境界の陰関数表現を入力した PointwiseMLP および GNN による予測結果を講演にて報告し、その有効性について議論する予定である。

4. 結論

本研究では、任意形状・自由度に対して汎用性を持つ機械学習モデルである、PointwiseMLP, GNN の温度場予測における予測精度を比較した。検証の結果、すべての検証データセットにおいて GNN が PointwiseMLP を上回る予測精度を示した。一方で、両モデルとも境界付近

における予測精度の悪化が観察された。この問題に対処するため、本研究では粒子数密度を用いた境界の陰関数表現を両モデルに導入し、境界付近の予測精度改善を図った。

謝辞: 本研究は、JSPS 科研費 23K24857 の支援を受けたものである。ここに記して謝意を表する。

参考文献

- [1] R. Grzeszczuk, D. Terzopoulos, and G. Hinton. Neuroanimator: fast neural network emulation and control of physics-based models. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 9–20, 1998.
- [2] L. Ladický, S. Jeong, B. Solenthaler, M. Pollefeys, and M. Gross. Data-driven fluid simulations using regression forests. *ACM Transactions on Graphics*, Vol. 34, No. 6, pp. 1–9, 2015.
- [3] S. He, Y. Li, Y. Feng, S. Ho, S. Ravanbakhsh, W. Chen, and B. Póczos. Learning to predict the cosmological structure formation. *Proceedings of the National Academy of Sciences*, Vol. 116, No. 28, pp. 13825–13832, 2019.
- [4] S. Wiewel, M. Becher, and N. Thuerey. Latent space physics: towards learning the temporal evolution of fluid flow. In *Computer Graphics Forum*, Vol. 38, pp. 71–82, Wiley Online Library, 2019.
- [5] M. Horie, N. Morita, T. Hishinuma, Y. Ihara, and N. Mitsume. Physics-embedded neural networks: graph neural PDE solvers with mixed boundary conditions. *Neural Information Processing Systems*, 2022.
- [6] I. I. Baskin, V. A. Palyulin, and N. S. Zefirov. A neural device for searching direct correlations between structures and properties of chemical compounds. *Journal of Chemical Information and Computer Sciences*, Vol. 37, No. 4, pp. 715–721, 1997.
- [7] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural Message Passing for Quantum Chemistry. *Proceedings of the 34th International Conference on Machine Learning*, PMLR 70:1263–1272, 2017.
- [8] F. Alet, A. K. Jeejawe, M. Bauza Villalonga, A. Rodriguez, T. Lozano-Perez, and L. Kaelbling. Graph element networks: adaptive, structured computation and memory. In *ICML*, 2019.
- [9] Á. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia. Learning to Simulate Complex Physics with Graph Networks. *Proceedings of the 37th International Conference on Machine Learning*, PMLR 119:8459–8468, 2020.
- [10] M. Horie. E(n)-Equivariant Graph Neural Networks Emulating Mesh-Discretized Physics. 筑波大学, 博士論文, 2023.
- [11] A. Paszke, S. Gross, S. Chintala, G. Chanan. "PyTorch". 2024-01, <https://github.com/pytorch/pytorch>, (参照 2024-02-05).