PRACTICAL IMPLICATIONS OF EQUIVARIANT AND INVARIANT GRAPH NEURAL NETWORKS FOR FLUID FLOW MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph neural networks (GNNs) have shown promise in learning unstructured mesh-based simulations of physical systems, including fluid dynamics. In tandem, geometric deep learning principles have informed the development of equivariant architectures. However, the practical implications of rotational equivariance in modeling fluids remains under-explored. We build a multi-scale equivariant GNN to forecast buoyancy-driven shear fluid flow and study the effect of modeling invariant and non-invariant representations of the flow state. Our results show that modeling invariant quantities produces more accurate long-term predictions and that these invariant quantities may be learned from the velocity field using a data-driven encoder.

1 INTRODUCTION

AI in the sciences is undergoing a renaissance of innovation as researchers are seeking to harness the potential of rapidly evolving deep learning methods. These methods are especially promising for applications such as approximating solutions to partial differential equations (PDEs) (Lu et al., 2021; Kovachki et al., 2021; Chen et al., 2018) because they can be made computationally efficient and data-driven, requiring partial domain knowledge to approximate solutions. Computational fluid dynamics (CFD) methods stand to gain significantly from AI-enhanced algorithms as more complex scientific and engineering flows test the limits of current numerical approaches (Brunton et al., 2020). Data-driven modeling has deep roots in CFD, but machine learning (ML) has provided new opportunities for model design, and many popular deep learning approaches have been explored for fluids applications in recent years.

Despite the proliferation and diversity of ML modeling approaches for fluids, generalization remains a core issue for data-driven models. Imbuing models with physically sound inductive biases has been identified as a potential path to better generalization by constraining the model class to satisfy certain physical laws, assumptions or behavior necessary for accurate predictions (Brunton, 2021). As fluid systems exhibit Euclidean symmetry, in this work we are concerned with the use of spatial and rotationally equivariant graph neural network (GNN) models. We examine four model architectures to understand the effect of embedded equivariance on modeling invariant and tensor-valued representations of fluid data. We assess model performance with regards to both accuracy and computational cost to provide a full picture of modeling strategies. We find that invariant representations of the flow state are effective for long-term forecasts of the flow field. Absent existing invariant representations, a neural network can be used to encode the invariant representation.

2 BACKGROUND AND RELATED WORK

Graph neural networks: Graph neural networks have gained traction in the deep learning community for applications where the data contain some underlying graph or network structure. GNNs are more flexible than convolutional neural networks (CNNs), which require the use of structured grid data, but still allow for some measure of locality through graph edges. Graph-based architectures have found a natural application in data-driven modeling of fluid flows, where data are typically obtained from numerical PDE solutions on unstructured computational meshes. The grid-independent framework is essential for problems in complex domains where a structured grid representation would require interpolation or other transformations. Indeed, GNNs have been used in a variety of fluid modeling contexts. Graph representations have been leveraged for modeling Lagrangian dynamics (Li et al., 2018; Ummenhofer et al., 2020; Sanchez-Gonzalez et al., 2020), steady-state predictions have been tackled in various works (Yang et al., 2022), including with solvers-in-the-loop (De Avila Belbute-Peres et al., 2020), and attention-style mechanisms have shown improvements in turbulence modeling (Peng et al., 2022). Pfaff et al. (2020) lays the foundation for unsteady PDE forecasting on graphs with derivative efforts (Lino et al., 2022) employing similar Encode-Process-Decode architectures.

Equivariance: We recognize Euclidean rotations and translations as the set of symmetries to embed in our models. While there are other important symmetries in fluid motion, such as Galilean invariance, we do not consider them here. Given a roto-translation operator R, an equivariant network f_θ obeys the following:

$$Rf_\theta(x) = f_\theta(Rx). \tag{1}$$

Equivariant neural networks are increasingly finding use in scientific machine learning applications. Rotation equivariant graph networks and tensor-field networks in particular have demonstrated value in accelerating molecular and N-body particle dynamics simulations (Thomas et al., 2018; Batzner et al., 2022). However, exploration of equivariance's role and application of these neural architectures to fluids problems has remained sparse in literature. Nevertheless, some studies leveraging equivariance and invariances for modeling fluid problems have been conducted. Ling et al. (2016) and Gao et al. (2020) use rotation invariant and equivariant networks for turbulence modeling. Rotation equivariant CNNs (Wang et al., 2020) and GNNs (Lino et al., 2022; Suk et al., 2022) have also demonstrated improved predictive accuracy in forecasting tasks.

3 METHODS

The model follows similar design principles as existing literature on learning simulations on graphs, such as Pfaff et al. (2020). The system is described at a point in time by a point cloud given by coordinates $\mathbf{x}_i \in \mathbf{x}_0, ..., \mathbf{x}_N$, fixed node features $\mathbf{f}_i$ corresponding to boundary conditions, external force fields, or global flow characteristics, e.g. Reynolds number, and modeled quantities $\mathbf{u}_i$, such as velocity and temperature. The model predicts the state of the system at the next time step $t+1$ and any future time steps $t+n$ using an iterative integrator. The model uses the Encode-Process-Decode paradigm, illustrated in Fig. 1, that has shown to be effective for modeling dynamical systems. The encoder generates the graph and transforms inputs into latent features. The latent features are processed with a series of message-passing layers and decoded to produce the update quantity passed to the integrator. We evaluate several model architectures that follow this general framework, described subsequently.

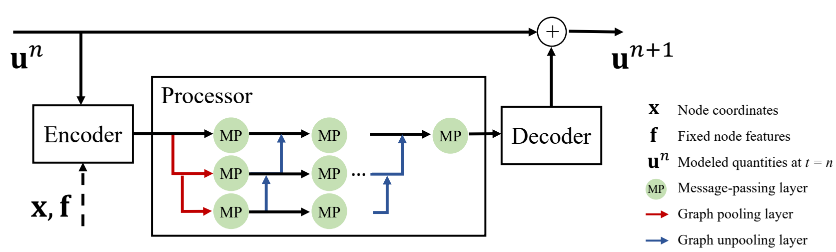

Figure 1: Schematic of the overall model architecture, indicating encoder and decoder blocks enclosing the multi-level message-passing graph processor.

Message-passing layers: Nonlinear graph message-passing layers comprise the bulk of the model, especially within the processor. Following general GNN schemes, an edge update is performed before aggregating edge features and applying a node update. Equivariance is achieved by ensuring that both updates are equivariant transformations. We leverage two equivariant operations, equivariant linear transformations and tensor products. Multi-layer perceptrons (MLPs) with one hidden layer are used; an MLP with equivariant linear layers is denoted MLP_{eq}.

Given incoming edge features $\mathbf{h}_{ij}$ and node features $\mathbf{h}_i$, first the edge update is computed:

$$\mathbf{v}_{ij} = \text{MLP}_{eq}(\mathbf{h}_{ij}, \mathbf{h}_i, \mathbf{h}_j) \tag{2}$$

$$\mathbf{h}'_{ij} = \mathbf{v}_{ij} \otimes (\mathbf{W})\, Y(\mathbf{r}_{ij}), \mathbf{W} = \text{MLP}(||\mathbf{r}_{ij}||), \tag{3}$$

where $\otimes(\mathbf{W})$ denotes a fully connected tensor product with linear weights $\mathbf{W}$, $\mathbf{r}_{ij}$ is the relative position vector $\mathbf{x}_i - \mathbf{x}_j$, $Y(\mathbf{r}_{ij})$ are spherical harmonics, and the edge update is given by $\mathbf{h}'_{ij}$, added as a residual update if the input and output sizes are the same. The node update is another MLP:

$$\mathbf{h}'_i = \text{MLP}_{eq}(\sum_{\mathcal{N}} \mathbf{h}'_{ij}, \mathbf{h}_i). \tag{4}$$

As we wish to explore the use of equivariance, we also form a non-equivariant layer. Here, any MLP_{eq}'s are replaced with MLPs. In addition, the tensor product reduces to a standard linear operation with weights $\mathbf{W}$. However, since any directional information from $\mathbf{r}_{ij}$ is lost, we include $\mathbf{x}_i, \mathbf{x}_j$ as additional inputs to the weight MLP. Finally, we utilize one more variation of the layer, the isotropic layer. This is nearly equivalent to the non-equivariant layer, except $||\mathbf{r}_{ij}||$ is again the only input to the weight MLP, producing an isotropic kernel. If the incoming edge and node features are genuine scalars, this is a rotationally equivariant operation.

Encoder/Decoder: The encoder transforms all inputs into both latent edge and latent node features for processing. Only nodal information $\mathbf{x}_i, \mathbf{f}_i, \mathbf{u}_i$ is provided as input. Thus, first edges with relative position vector $\mathbf{r}_{ij}$ are generated using a radial cutoff. Two steps complete the transformation – a node- and edge-wise linear projection to the hidden layer and one message-passing layer to the output latent space. The encoder can also be made equivariant or non-equivariant by selecting the appropriate linear and message-passing layers. The decoder mirrors the encoder architecture, however, only the node features are decoded. The output is passed to the integrator to produce the system state at the next time step.

Graph coarsening: Research efforts on graph representation learning have motivated the development of various graph coarsening or pooling strategies (Mesquita et al., 2020; Chen et al., 2022; Gao & Ji, 2022). When data are represented on a physical grid in space, pooling can capture multi-scale features that have been shown to be effective for better convergence and accuracy of models. We use a very simple coarsening strategy in the model for improved learning by generating a random subgraph using uniform sampling of the top-level graph nodes and a radial cutoff (specific to the graph level) to produce edges. The unpooling operation distributes lower-level node features to an empty feature matrix on the top-level graph, while a node-wise MLP computes new top-level features.

Models and training: Various combinations of the encoder/decoder and processor blocks are combined to produce three distinct model classes. The first equivariant model, denoted as **eq**, encodes latent features comprising scalar and vector-valued quantities using the equivariant encoder/decoder and equivariant processor. Another equivariant model (**eqscl**) generates scalar invariant latent features using the equivariant encoder, but the processor consists of isotropic message-passing layers, which provides a computational advantage. The non-equivariant model (**neq**) trades all equivariant blocks for non-equivariant blocks. A fourth model is tested, **neqaug**, which uses the same architecture as **neq**, but equivariance is included by augmenting the training data with random rotations.

Models are constructed in PyTorch using the `e3nn` package (Geiger et al., 2022) for equivariant operations and `PyG` (Fey & Lenssen, 2019) for graph-specific functionality. Models are trained with the Adam optimizer using a decaying learning rate varying from 10^{-3} to 10^{-5} on 8 V100 GPUs. To predict long trajectories, the standard approach of perturbing training samples with noise is used.

4 EXPERIMENTS

We evaluate our approach on a canonical strong-shear flow that exhibits the Kelvin-Helmholtz instability, known as Marsigli flow (Ahmed et al., 2021). Two fluids of different densities, or equivalently temperatures, are contained in a channel separated by a barrier. Once the barrier is removed, the fluids mix according to the Boussinesq equations. Marsigli flow is often used as an idealized case study for understanding ocean current dynamics. This is a particularly challenging problem for data-driven models due to the highly transient nature of the flow. Numerical ground truth solutions are obtained using standard second-order central finite difference schemes. The training dataset contains flow at four different Re, 700, 900, 1100, and 1300 for 8 seconds. The test set consists of flow at $Re = 1000$.

Data type	Model type	1-step test MSE	Forecast R^2	Equivariance MSE	Train time (hrs)	Evaluation time (s)
wst	**neq**	7.55e-7	0.9988	4.88e-3	12.9	3.41
wst	**neqaug**	2.25e-6	0.9324	1.27e-6	19.5	3.40
wst	**eq**	2.60e-6	0.9700	1.02e-7	64.2	9.38
wst	**eqscl**	1.86e-6	0.9972	1.32e-7	19.3	3.94
uvt	**neq**	7.58e-7	0.9772	1.40e-3	12.9	3.71
uvt	**neqaug**	2.57e-6	0.9487	8.41e-7	19.5	3.68
uvt	**eq**	1.84e-6	0.9755	1.42e-7	65.2	9.07
uvt	**eqscl**	7.75e-7	0.9863	9.20e-7	19.5	4.10

Table 1: Comparison of model performance on the Marsigli flow test dataset.

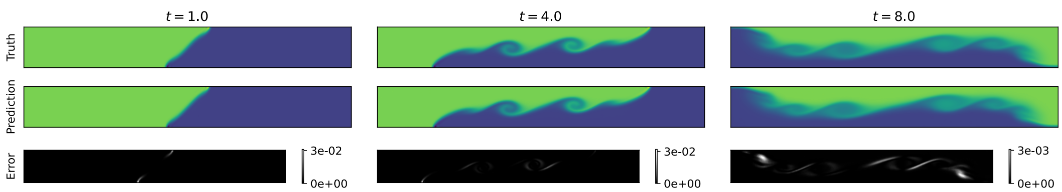

Figure 2: Snapshots of equivariant model **wst-eqscl** temperature field predictions at an unseen Reynolds number compared to the ground truth.

The fluid state can be represented by the temperature field and either the velocity field or the vorticity and streamfunction. The vorticity and streamfunction are scalar invariant fields in two-dimensions and thus do not transform under rotation, in contrast to the velocity field. We investigate the effect of modeling both representations in addition to model architecture choices. The data are coded with tags **wst** or **uvt** for modeling vorticity, streamfunction, temperature or velocity and temperature respectively.

Fig. 2 shows a qualitative comparison of temperature fields from the ground truth simulation and model **wst-eqscl**. In Table 1, we report the 1-step mean-squared error (MSE) on the test set, the coefficient of determination R^2 on the full trajectory forecast, an unsupervised equivariance error comparing both sides of Eq. 1, as well as train and evaluation times of each model. Metrics are computed using the temperature field, which is consistent across flow state representations. All models achieve comparable 1-step errors, however, differences are observed in the forecast R^2. Generally, modeling the invariant fields **wst** results in higher forecast accuracy, except for within the data-augmented models. Data-augmentation also leads to a significant decrease in accuracy, although equivariance can be maintained. **eqscl** consistently outperforms **eq** in both cost and forecast accuracy, which we believe is due to the invariant latent state representation instead of higher-order tensor representations. While it may be expected that **neq** would trade higher accuracy for higher equivariance error, we find that when modeling **uvt**, **eqscl** improves over both baselines **neq** and **neqaug** and maintains equivariance at a reasonable computational cost.

5 Conclusions

We assess the efficacy of several rotation equivariant and non-equivariant graph neural networks on modeling spatiotemporal dynamics exhibited by buoyancy-driven fluid flow. The models are trained on both invariant and non-invariant representations of the flow state and tested by forecasting the flow field at an unseen Reynolds number. We see that the use of invariant representations provides a significant benefit to the generalization task. Embedded equivariance is most effective when modeling the vector-valued velocity field, and can be made computationally efficient with invariant latent representations. In fluid forecasting tasks, we suggest modeling invariant quantities, with data-driven invariant encoders serving as a viable alternative if invariant representations are unknown.

REFERENCES

Shady E. Ahmed, Omer San, Kursat Kara, Rami Younis, and Adil Rasheed. Multifidelity computing for coupling full and reduced order models. *PLOS ONE*, 16(2):e0246092, February 2021. doi: 10.1371/journal.pone.0246092. URL https://doi.org/10.1371/journal.pone.0246092.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13 (1), May 2022. doi: 10.1038/s41467-022-29939-5. URL https://doi.org/10.1038/s41467-022-29939-5.

Steven L Brunton. Applying machine learning to study fluid mechanics. *Acta Mechanica Sinica*, 37 (12):1718–1726, 2021.

Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52(1):477–508, January 2020. doi: 10.1146/annurev-fluid-010719-060214. URL https://doi.org/10.1146/annurev-fluid-010719-060214.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.

Yuzhao Chen, Yatao Bian, Jiying Zhang, Xi Xiao, Tingyang Xv, and Yu Rong. Diversified multiscale graph learning with graph self-correction. In Alexander Cloninger, Timothy Doster, Tegan Emerson, Manohar Kaul, Ira Ktena, Henry Kvinge, Nina Miolane, Bastian Rieck, Sarah Tymochko, and Guy Wolf (eds.), *Proceedings of Topological, Algebraic, and Geometric Learning Workshops 2022*, volume 196 of *Proceedings of Machine Learning Research*, pp. 48–54. PMLR, 25 Feb–22 Jul 2022. URL https://proceedings.mlr.press/v196/chen22a.html.

Filipe De Avila Belbute-Peres, Thomas Economon, and Zico Kolter. Combining differentiable PDE solvers and graph neural networks for fluid flow prediction. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2402–2411. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/de-avila-belbute-peres20a.html.

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Hongyang Gao and Shuiwang Ji. Graph u-nets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4948–4960, 2022. doi: 10.1109/TPAMI.2021.3081010.

Liyao Gao, Yifan Du, Hongshan Li, and Guang Lin. Roteqnet: Rotation-equivariant network for fluid systems with symmetric high-order tensors. *CoRR*, abs/2005.04286, 2020. URL https://arxiv.org/abs/2005.04286.

Mario Geiger, Tess Smidt, Alby M., Benjamin Kurt Miller, Wouter Boomsma, Bradley Dice, Kostiantyn Lapchevskyi, Maurice Weiler, Michał Tyszkiewicz, Simon Batzner, Dylan Madisetti, Martin Uhrin, Jes Frellsen, Nuri Jung, Sophia Sanborn, Mingjian Wen, Josh Rackers, Marcel Rød, and Michael Bailey. Euclidean neural networks: e3nn, April 2022. URL https://doi.org/10.5281/zenodo.6459381.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces, 2021. URL https://arxiv.org/abs/2108.08481.

Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids, 2018. URL https://arxiv.org/abs/1810.01566.

Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016. doi: 10.1017/jfm.2016.615.

Mario Lino, Stathi Fotiadis, Anil A. Bharath, and Chris D. Cantwell. Multi-scale rotation-equivariant graph neural networks for unsteady eulerian fluid dynamics. *Physics of Fluids*, 34(8): 087110, 2022. doi: 10.1063/5.0097679. URL `https://doi.org/10.1063/5.0097679`.

Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, January 2021. doi: 10.1137/19m1274067. URL `https://doi.org/10.1137/19m1274067`.

D. Mesquita, A. H. Souza, and S. Kaski. Rethinking pooling in graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Wenhui Peng, Zelong Yuan, and Jianchun Wang. Attention-enhanced neural network models for turbulence simulation. *Physics of Fluids*, 34(2):025111, February 2022. doi: 10.1063/5.0079302. URL `https://doi.org/10.1063/5.0079302`.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. 2020. doi: 10.48550/ARXIV.2010.03409. URL `https://arxiv.org/abs/2010.03409`.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8459–8468. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/sanchez-gonzalez20a.html`.

Julian Suk, Pim de Haan, Phillip Lippe, Christoph Brune, and Jelmer M. Wolterink. Mesh convolutional neural networks for wall shear stress estimation in 3d artery models. In Esther Puyol Antón, Mihaela Pop, Carlos Martín-Isla, Maxime Sermesant, Avan Suinesiaputra, Oscar Camara, Karim Lekadir, and Alistair Young (eds.), *Statistical Atlases and Computational Models of the Heart. Multi-Disease, Multi-View, and Multi-Center Right Ventricular Segmentation in Cardiac MRI Challenge*, pp. 93–102, Cham, 2022. Springer International Publishing. ISBN 978-3-030-93722-5.

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, 2018. URL `https://arxiv.org/abs/1802.08219`.

Benjamin Ummenhofer, Lukas Prantl, Nils Thuerey, and Vladlen Koltun. Lagrangian fluid simulation with continuous convolutions. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=B1lDoJSYDH`.

Rui Wang, Robin Walters, and Rose Yu. Incorporating symmetry into deep dynamics models for improved generalization, 2020. URL `https://arxiv.org/abs/2002.03061`.

Zhishuang Yang, Yidao Dong, Xiaogang Deng, and Laiping Zhang. AMGNET: multi-scale graph neural networks for flow field prediction. *Connection Science*, 34(1):2500–2519, October 2022. doi: 10.1080/09540091.2022.2131737. URL `https://doi.org/10.1080/09540091.2022.2131737`.