
Journal of Scientific Computing and Machine Learning, vol. 1, pp. 43–51, 2025
Submission Date: January 15, 2025 Acceptance Date: January 26, 2025

Generalized Lie Symmetries in Physics-Informed Neural Operators

Amy Xiang Wang[1], Zakhar Shumaylov[2], Peter Zaika[2], Ferdia Sherry[2], and Carola-Bibiane Sch[2]

[1]New York University
[2]University of Cambridge
xw914@nyu.edu

Abstract

Physics-informed neural operators (PINOs) have emerged as powerful tools for learning solution operators of partial differential equations (PDEs). Recent research has demonstrated that incorporating Lie point symmetry information can significantly enhance the training efficiency of PINOs, primarily through techniques like data, architecture, and loss augmentation. In this work, we focus on the latter, highlighting that point symmetries oftentimes result in no training signal, limiting their effectiveness in many problems. To address this, we propose a novel loss augmentation strategy that leverages evolutionary representatives of point symmetries, a specific class of generalized symmetries of the underlying PDE. These generalized symmetries provide a richer set of generators compared to standard symmetries, leading to a more informative training signal. We demonstrate that leveraging evolutionary representatives enhances the performance of neural operators, resulting in improved data efficiency and accuracy during training.

Keywords Deep learning · Physics-informed · Neural Operator · PINN · Lie point symmetries · Generalized symmetries

1. Introduction

Deep neural networks are increasingly used for scientific computing, particularly in simulating complex physical systems governed by partial differential equations (PDEs) [11], where traditional methods struggle [24].

Physics-informed neural networks (PINNs) [24] integrate physical laws and constraints directly into the learning process, demonstrating success in various applications [24, 29, 35, 14, 27, 33]. However, classical PINNs face challenges under varying parameters and boundary conditions and often encounter training challenges due to unbalanced gradients and non-convex loss landscapes [10, 15, 22]. Neural operators offer a compelling alternative, which unlike standard feedforward neural networks, learn mappings between infinite-dimensional function spaces, directly connecting parameters and initial/boundary conditions to PDE solutions [21, 32]. This eliminates the need for independent simulations and, when combined with physics-informed loss functions, often improves generalization and physical validity. We focus on this class of approaches.

In a separate direction, the field of *geometric deep learning* has demonstrated the practical and theoretical benefits of incorporating symmetry information into neural networks, leading to improved performance through beneficial inductive biases [6, 5]. For example,

the translational equivariance of convolutional neural networks (CNNs) [9, 18] has been suggested as an important reason for their successes in tasks such as image classification [16, 12].

Similarly, recent work in neural operators has shown that incorporating PDE symmetries improves generalization and training efficiency. However, most research has focused on Lie point symmetries due to their well-established theoretical foundation and systematic derivation methods [23, 13, 3]. In contrast, the potential of generalized symmetries remains largely unexplored due to their unsuitability for data augmentation or equivariant architectures.

This paper demonstrates that generalized PDE symmetries can nonetheless provide valuable inductive biases when incorporated through loss augmentation.

1.1. PDE Symmetries in Deep Learning

Incorporation of point symmetries into neural solvers has received significant attention, leading to the development of many approaches that can be broadly classified into three categories:

Data augmentation is the simplest way to add symmetry information into neural PDE solvers [4, 20] by augmenting the training data with transformed versions obtained by applying symmetry operations. While straightforward to implement, this approach can increase training costs.

Loss augmentation is an alternative way to inject symmetry information by appropriately regularizing the loss using symmetry information [1, 19, 34]. This often leads to improved generalization and sample efficiency, although it may not guarantee equivariant models.

Architecture augmentation aims to directly embed symmetries into the model architecture. Given the complexity of point symmetry groups associated with PDEs, this requires either general techniques [26] or exploiting simpler subgroups with existing architectures [31].

Beyond these, alternative approaches exist, based on problem reformulations [2, 17] or those focused on other types of structures, e.g. from Hamiltonian dynamics [7, 8, 30].

2. PDE Symmetries

Here, we introduce the notion of PDE symmetries. The formal presentation here closely follows [23], and the reader is strongly encouraged to read it for more in-depth discussions of the concepts.

In simple words, PDE point symmetries are transformations that map solutions of a given PDE to other solutions of the same PDE. To derive these symmetries, traditionally the focus is shifted to Jet spaces, where the problem of studying PDE symmetries is transformed into the simpler task of studying symmetries of algebraic equations. By employing one-parameter groups of transformations, it becomes possible to systematically derive all such symmetries by examining their infinitesimal actions.

Suppose we are considering a system Δ of n-th order differential equations involving p independent $x = \left(x^1, \dots, x^p\right) \in X$, and q dependent variables $u = \left(u^1, \dots, u^q\right) \in U$.

Jet Spaces and Prolongations: Consider $f : X \to U$ smooth and let U_k be the Euclidean space, endowed with coordinates $u^\alpha_J = \partial_J f^\alpha(x)$ in multi-index notation so as to represent the above derivatives. Furthermore, set $U^{(n)} = U \times U_1 \times \cdots \times U_n$ to be the Cartesian product space, whose coordinates represent all the derivatives of functions $u = f(x)$ of all orders from 0 to n. The total space $X \times U^{(n)}$, whose coordinates represent the independent variables, the dependent variables and the derivatives of the dependent variables up to order n is called the n-th order jet space of the underlying space $X \times U$. Given a smooth function $u = f(x)$, there is an induced function $u^{(n)} = \mathrm{pr}^{(n)} f(x)$, called the n-th prolongation of f, defined by the equations $u^\alpha_J = \partial_J f^\alpha(x)$. Thus $\mathrm{pr}^{(n)} f$ is a function from X to the space $U^{(n)}$, and for each x in X, $\mathrm{pr}^{(n)} f(x)$ is a vector representing values of f and all its derivatives up to order n at the point x.

The PDE is a system of equations $\Delta\left(x, u^{(n)}\right) = 0$, with $\Delta : X \times U^{(n)} \to \mathbb{R}^l$ smooth, determining a subvariety

$$S_\Delta = \left\{ \left(x, u^{(n)}\right) : \Delta\left(x, u^{(n)}\right) = 0 \right\} \subset X \times U^{(n)}$$

A symmetry group of the system Δ will be a local group of transformations, G^Δ, acting on some open

subset $M \subset X \times U$ such that "G transforms solutions of Δ to other solutions of Δ", i.e. leaving S_Δ invariant.

Prolongations of actions Now suppose G acts on an open $M \subset X \times U$, i.e. independent and dependent variables. There is an induced local action of G on the n-jet space $M^{(n)}$, called the n-th prolongation of G denoted $\mathrm{pr}^{(n)}G$. This prolongation is defined so that it transforms the derivatives of functions $u = f(x)$ into the corresponding derivatives of the transformed function $\tilde{u} = \tilde{f}(\tilde{x})$. Consider now v a vector field on M, with a corresponding one-parameter group $\exp(\varepsilon v)$. The n-th prolongation of v is denoted $\mathrm{pr}^{(n)}\, v$, will be a vector field on the n-jet space $M^{(n)}$, and is defined to be the infinitesimal generator of the corresponding prolonged one-parameter group $\mathrm{pr}^{(n)}[\exp(\varepsilon v)]$. Relevance of these concepts is highlighted by the following theorem:

Theorem 2.1 (2.31 in [23]). *Suppose $\Delta\left(x, u^{(n)}\right) = 0$ is a system of differential equations of maximal rank defined over $M \subset X \times U$. If G is a local group of transformations acting on M, and*

$$\mathrm{pr}^{(n)}\, v\left[\Delta\left(x, u^{(n)}\right)\right] = 0, \quad \textit{whenever } \Delta\left(x, u^{(n)}\right) = 0$$

for every infinitesimal generator v of G, then G is a symmetry group of the system.

Generalized Symmetries We can consider some vector field acting on M of the form

$$v = \sum_{i=1}^{p} \xi^i(x,u)\frac{\partial}{\partial x^i} + \sum_{\alpha=1}^{q} \phi_\alpha(x,u)\frac{\partial}{\partial u^\alpha},$$

and provided the coefficient functions ξ^i, ϕ_α depend only on x and u, v generates a one-parameter group of transformations $\exp(\varepsilon v)$ acting pointwise. A significant generalization of the notion of symmetry group is obtained by relaxing this geometrical assumption, and allowing the coefficients ξ^i, ϕ_α to also depend on derivatives of u.

Definition 2.2 (Generalized Vector Fields)**.** A generalized vector field is an expression of the form

$$v = \sum_{i=1}^{p} \xi^i[u]\frac{\partial}{\partial x^i} + \sum_{\alpha=1}^{q} \phi_\alpha[u]\frac{\partial}{\partial u^\alpha}$$

in which ξ^i and ϕ_α are smooth differential functions.

Example 2.3. An example of a generalized vector field is $v = u_x\partial_u$, which admits the following prolongation:

$$\mathrm{pr}\, v = u_x\frac{\partial}{\partial u} + u_{xx}\frac{\partial}{\partial u_x} + u_{xt}\frac{\partial}{\partial u_t} + u_{xxx}\frac{\partial}{\partial u_{xx}} + \cdots$$

and thus turns out to be a generalized symmetry of the Burgers' Equation (6), as $\mathrm{pr}\, v\,[\Delta_B] = 0$.

Evolutionary Vector Fields Among all the generalized vector fields, those in which the coefficients $\xi^i[u]$ of the $\partial/\partial x^i$ are zero play a particularly important role.

Definition 2.4 (5.4 in [23])**.** Let $Q[u] = (Q_1[u], \ldots, Q_q[u])$ be a q-tuple of differential functions. The generalized vector field

$$v_Q = \sum_{\alpha=1}^{q} Q_\alpha[u]\frac{\partial}{\partial u^\alpha}$$

is called an evolutionary vector field, and Q is called its characteristic.

Any generalized vector field v has an associated evolutionary representative v_Q with characteristic $Q_\alpha = \phi_\alpha - \sum_{i=1}^{p} \xi^i \partial u^\alpha/\partial x^i$, determining essentially the same symmetry, as illustrated by the following proposition:

Proposition 2.5 (5.5 in [23])**.** *A generalized vector field v is a symmetry of a system of differential equations if and only if its evolutionary representative v_Q is.*

Evolutionary vector fields are particularly appealing as they capture the essence of the symmetry, discarding the dependent variable information, while also admitting much simpler expressions for their prolongations.

However, it is important to note that these are not symmetries in the traditional sense, as they no longer act on $X \times U$. For a more in-depth discussion, see [23].

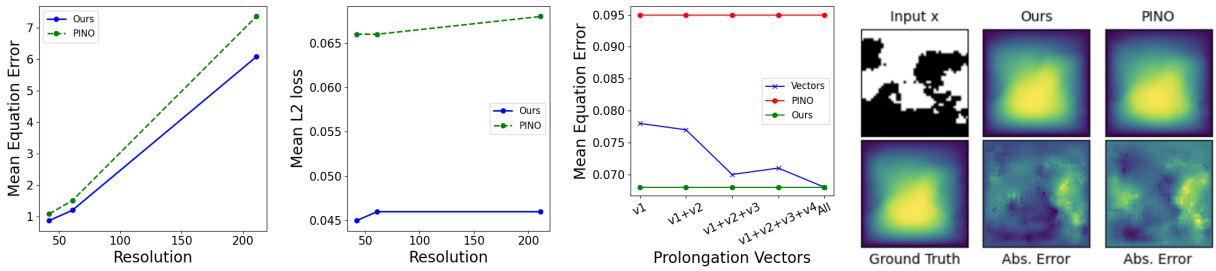

Figure 1: **Left** (first two): Darcy Flow zero-shot evaluation by resolution. **Middle**: Burgers' equation symmetry regularization effectiveness by vector. **Right**: Darcy Flow prediction results comparison with 100 training samples

Table 1: Comparison of L_2 loss and equation error for Darcy Flow and Burgers' equation with varying sample sizes, using our symmetry-based method, PINO (no symmetry), and the method from [1, 34].

Equation	No. Samples	Metric	Symmetry (Ours)	No Symmetry (PINO)	Symmetry [1, 34]
Darcy Flow	100	L_2 Loss	$\mathbf{0.046 \pm 1e-3}$	$0.066 \pm 1e-2$	
	250		$\mathbf{0.029 \pm 7e-4}$	$0.323 \pm 8e-4$	
	500		$\mathbf{0.014 \pm 3e-4}$	$0.017 \pm 4e-4$	
	1000		$\mathbf{0.011 \pm 2e-3}$	$0.013 \pm 3e-4$	
	100	Eqn. Error	$\mathbf{1.202 \pm 1e-2}$	$1.551 \pm 1e-2$	
	250		$\mathbf{0.621 \pm 7e-3}$	$0.925 \pm 1e-2$	
	500		$\mathbf{0.579 \pm 7e-3}$	$0.723 \pm 9e-3$	
	1000		$\mathbf{0.491 \pm 6e-3}$	$0.583 \pm 7e-3$	
Burgers' Eqn.	25	L_2 Loss	$\mathbf{0.082 \pm 6e-3}$	$0.089 \pm 6e-3$	$0.099 \pm 6e-3$
	50		$\mathbf{0.044 \pm 3e-3}$	$\mathbf{0.044 \pm 3e-2}$	$0.050 \pm 4e-3$
	100		$\mathbf{0.018 \pm 1e-3}$	$0.019 \pm 1e-3$	$0.022 \pm 1e-3$
	25	Eqn. Error	$\mathbf{0.182 \pm 3e-2}$	$0.217 \pm 4e-2$	$0.440 \pm 6e-2$
	50		$\mathbf{0.068 \pm 1e-2}$	$0.095 \pm 2e-2$	$0.217 \pm 4e-2$
	100		$\mathbf{0.023 \pm 4e-3}$	$0.024 \pm 4e-3$	$0.068 \pm 1e-2$

3. Methodology

Based on the discussion in Section 2, instead of considering symmetry vector fields, we propose to use evolutionary representatives of these vector fields instead. These turn out to be more informative in practice, as illustrated below.

To understand the proposed methodology, we first recap in simpler notation, the method of [1, 34]. Clasically, PINNs aim to solve the PDE by minimizing the residual

$$\min_{\theta} \ \|\Delta[u_\theta]\|_2^2. \tag{1}$$

In [1, 34], it is proposed that for a Lie algebra v_i, it is beneficial to minimize (for some $\gamma > 0$):

$$\min_{\theta} \ \|\Delta[u_\theta]\|_2^2 + \gamma \sum_i \|\mathrm{pr}\, v_i\, [\Delta]\, [u_\theta]\|_2^2, \tag{2}$$

as by Theorem 2.1, zero loss solutions of Equation (1) are also zero loss solutions of Equation (2). Based on Theorem 2.5, we instead propose to use the evolutionary representatives of v_i, via

$$\min_{\theta} \ \|\Delta[u_\theta]\|_2^2 + \gamma \sum_i \left\|\mathrm{pr}\, [v_i]_Q\, [\Delta]\, [u_\theta]\right\|_2^2, \tag{3}$$

as motivated by the following example:

Example 3.1 (Burgers). Consider Burgers' Equation (6), and consider the symmetry generated by $v_1 = \partial_x$. Its evolutionary representative, as above, is $[v_1]_Q = -u_x \partial_u$. We can derive both prolongation actions as:

$$\mathrm{pr}\, v_1\, [\Delta_B] = 0, \quad \mathrm{pr}\, [v_1]_Q\, [\Delta_B] = -D_x\, [\Delta_B], \tag{4}$$

where D_x denotes the total x derivative. This shows that standard point symmetries result in no extra

terms in Equation (2), unlike Equation (3). Same occurs for all other generators for both Burgers' equation (Appendix A) and Darcy Flow (Appendix B).

4. Experiments

In this section we present numerical results showcasing the utility of generalized symmetries for training PINOs. This is illustrated for the 2D Darcy flow and the 1D Burgers' equation. Our code is publicly available at `https://github.com/xiwang129/GPS_PINO`.

4.1. Darcy Flow

We consider solving the 2D Darcy Flow equation

$$\Delta_D := \nabla \cdot (k(x)\nabla u(x)) + f(x) = 0, \tag{5}$$

on the domain $x \in [0,1]^2$ with $u(x) = 0$ on the boundary, where $k(x)$ is the permeability field and $f(x)$ is the source term. We tested our approach on a 61×61 resolution dataset downsampled from a 421×421 resolution dataset and set $f(x) = 1$ as in [21]. We trained a 2D Fourier neural operator model for 300 epochs using the proposed generalized symmetry loss in Equation (3). As is common with PINOs, in this example we also include a data loss term enforcing initial and boundary conditions, and a trajectory matching loss.The trained model is tested on 500 samples. Table 1 indicates that the proposed symmetry regularization improves prediction in terms of both L_2 and equation losses, being more sample efficient in minimizing the PDE residuals. As shown in Equation (10), the point symmetry method of [1, 34] in Equation (2) generates no extra terms, being equivalent to a standard PINO.

Moreover, we experimented with zero-shot prediction by testing the trained 61×61 model on other resolutions: 40×40 and 211×211. Figure 1 left illustrates that the resulting model consistently outperforms the baselines in both L_2 and equation loss.

4.2. Burgers' equation

We consider solving the 1D Burgers' equation

$$\Delta_B := u_t + uu_x - \nu u_{xx} = 0, \tag{6}$$

on the domain $x \in [0,1]$ and $t \in [0,1]$ with initial condition $u(x,0) = u_0(x)$, for which we generated the 1D Burgers' dataset following [25], using a 128×100 grid and setting the viscosity coefficient to $\nu = 0.01$. We compared the proposed method in Equation (3) with [1, 34] defined in Equation (2). Table 1 presents the results, demonstrating that the proposed method consistently achieves either superior or comparable performance to the baselines with respect to both equation error and L_2 error. Following experimental details of [1], the residual for this equation does not appear directly in the training objective. Figure 2 provides a representative prediction showcasing the improved error achieved by our method.

We further evaluated the impact of each of the Lie algebra vectors from Equation (7) on the resulting equation error, by adding one term at a time. An example is illustrated in the middle pane of Figure 1, showcasing that the introduction of even a single Lie algebra vector yields a substantial reduction in the equation error, while the benefit of incorporating additional vectors becomes marginal.

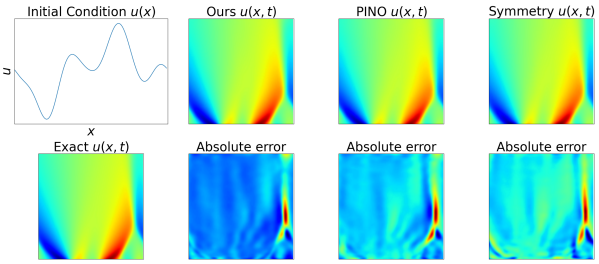

Figure 2: Visual comparison of trajectory predictions for Burgers' equation trained with 50 samples.

5. Conclusion

This work demonstrates that evolutionary representatives of point symmetries enhance data efficiency in training PINOs, resulting in a useful training signal even when the underlying point symmetries themselves can not be used.

While this study focused on point symmetries, the framework presented is readily applicable to any generalized symmetry. We hypothesize that incorporating such symmetries can similarly improve trainability and we leave a proper evaluation of this to future work.

We further posit that the observed efficiency partly arises from the inducement of a Sobolev-type norm [28]. Crucially, our approach offers a systematic way for selecting the appropriate norm based on the governing PDE. This connection between generalized symmetries, Sobolev norms, and improved training warrants further investigation.

Acknowledgments

CBS acknowledges support from the Philip Leverhulme Prize, UK, the Royal Society Wolfson Fellowship, UK, the EPSRC advanced career fellowship, UK EP/V029428/1, EPSRC grants EP/S026045/1 and EP/T003553/1, EP/N014588/1, EP/T017961/1, the Wellcome Innovator Awards, UK 215733/Z/19/Z and 221633/Z/20/Z, the European Union Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No.777826 NoMADS, the Cantab Capital Institute for the Mathematics of Information, UK and the Alan Turing Institute, UK. FS acknowledges support from the EPSRC advanced career fellowship EP/V029428/1. ZS acknowledges support from the Cantab Capital Institute for the Mathematics of Information, Christs College and the Trinity Henry Barlow Scholarship scheme. AXW acknowledges the support from NYU HPC.

References

[1] Tara Akhound-Sadegh, Laurence Perreault-Levasseur, Johannes Brandstetter, Max Welling, and Siamak Ravanbakhsh. Lie point symmetry and physics-informed networks. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 42468–42481. Curran Associates, Inc., 2023.

[2] Shivam Arora, Alex Bihlo, and Francis Valiquette. Invariant physics-informed neural networks for ordinary differential equations, 2024. arXiv:2310.17053.

[3] Gerd Baumann. MathLie a program of doing symmetry analysis. *Mathematics and Computers in Simulation*, 48:205–223, 1998.

[4] Johannes Brandstetter, Max Welling, and Daniel E Worrall. Lie point symmetry data augmentation for neural PDE solvers. In *International Conference on Machine Learning*, pages 2241–2256. PMLR, 2022.

[5] Johann Brehmer, Sönke Behrends, Pim de Haan, and Taco Cohen. Does equivariance matter at scale?, 2024.

[6] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Velickovic. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *CoRR*, abs/2104.13478, 2021.

[7] Priscilla Canizares, Davide Murari, Carola-Bibiane Schönlieb, Ferdia Sherry, and Zakhar Shumaylov. Hamiltonian matching for symplectic neural integrators. *arXiv preprint arXiv:2410.18262*, 2024.

[8] Priscilla Canizares, Davide Murari, Carola-Bibiane Schönlieb, Ferdia Sherry, and Zakhar Shumaylov. Symplectic neural flows for modeling and discovery. *arXiv preprint arXiv:2412.16787*, 2024.

[9] Kunihiko Fukushima and Sei Miyake. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, 15(6):455–469, 1982.

[10] Tamara G Grossmann, Urszula Julia Komorowska, Jonas Latz, and Carola-Bibiane Schönlieb. Can physics-informed neural networks beat the finite element method? *arXiv preprint arXiv:2302.04107*, 2023.

[11] Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[13] Nail H Ibragimov. *CRC handbook of Lie group analysis of differential equations*, volume 3. CRC press, 1995.

[14] Sharmila Karumuri, Rohit Tripathy, Ilias Bilionis, and Jitesh Panchal. Simulator-free solution of high-dimensional stochastic elliptic partial differential equations using deep neural networks. *Journal of Computational Physics*, 404:109120, 2020.

[15] Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems*, 34:26548–26560, 2021.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[17] Pierre-Yves Lagrave and Eliot Tron. Equivariant neural networks and differential invariants theory for solving partial differential equations. In *Physical Sciences Forum*, volume 5, page 13. MDPI, 2022.

[18] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[19] Jie-Ying Li, Hui Zhang, Ye Liu, Lei-Lei Guo, Li-Sheng Zhang, and Zhi-Yong Zhang. Utilizing symmetry-enhanced physics-informed neural network to obtain the solution beyond sampling domain for partial differential equations. *Nonlinear Dynamics*, 111(23):21861–21876, 2023.

[20] Ye Li, Yiwen Pang, and Bin Shan. Physics-guided data augmentation for learning the solution operator of linear differential equations. In *2022 IEEE 8th International Conference on Cloud Computing and Intelligent Systems (CCIS)*, pages 543–547, 2022.

[21] Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM/JMS Journal of Data Science*, 2021.

[22] Johannes Müller and Marius Zeinhofer. Position: Optimization in SciML should employ the function space geometry. In *Forty-first International Conference on Machine Learning*, 2024.

[23] Peter J Olver. *Applications of Lie groups to differential equations*, volume 107. Springer Science & Business Media, 1993.

[24] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[25] Shawn G Rosofsky, Hani Al Majed, and EA Huerta. Applications of physics informed neural operators. *Machine Learning: Science and Technology*, 4(2):025022, 2023.

[26] Zakhar Shumaylov, Peter Zaika, James Rowbottom, Ferdia Sherry, Melanie Weber, and Carola-Bibiane Schönlieb. Lie algebra canonicalization: Equivariant neural operators under arbitrary lie groups. *arXiv preprint arXiv:2410.02698*, 2024.

[27] Justin Sirignano and Konstantinos Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018.

[28] Hwijae Son, Jin Woo Jang, Woo Jin Han, and Hyung Ju Hwang. Sobolev training for physics-informed neural networks. *Communications in Mathematical Sciences*, 21(6):1679–1705, 2023.

[29] Luning Sun, Han Gao, Shaowu Pan, and Jian-Xun Wang. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering*, 361:112732, 2020.

[30] Yusuke Tanaka, Takaharu Yaguchi, Tomoharu Iwata, and Naonori Ueda. Neural operators meet energy-based theory: Operator learning for hamiltonian and dissipative pdes, 2024.

[31] Rui Wang, Robin Walters, and Rose Yu. Incorporating symmetry into deep dynamics models for improved generalization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[32] Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Science advances*, 7(40):eabi8605, 2021.

[33] Lipei Zhang, Yanqi Cheng, Lihao Liu, Carola-Bibiane Schönlieb, and Angelica I Aviles-Rivero. Biophysics informed pathological regularisation for brain tumour segmentation. *arXiv preprint arXiv:2403.09136*, 2024.

[34] Zhi-Yong Zhang, Hui Zhang, Li-Sheng Zhang, and Lei-Lei Guo. Enforcing continuous symmetries in physics-informed neural network for solving forward and inverse problems of partial differential equations. *Journal of Computational Physics*, 492:112415, 2023.

[35] Yinhao Zhu, Nicholas Zabaras, Phaedon-Stelios Koutsourelakis, and Paris Perdikaris. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics*, 394:56–81, 2019.

A. Burgers' Equation Symmetries

In this appendix, we provide detailed derivations of the evolutionary representatives and their prolongations for the symmetries of Burgers' equation.

Burgers' equation is given by:

$$\Delta_B \coloneqq u_t + uu_x - \nu u_{xx} = 0$$

The Lie algebra of symmetries for Burgers' equation includes the following generators:

$$\begin{aligned} v_1 &= \partial_x, \\ v_2 &= \partial_t, \\ v_3 &= 2t\partial_t + x\partial_x - u\partial_u, \\ v_4 &= t\partial_x + \partial_u, \\ v_5 &= t^2\partial_t + tx\partial_x + (x - tu)\partial_u. \end{aligned}$$

It turns out to be rather simple to find evolutionary representatives of these as:

$$\begin{aligned} [v_1]_Q &= -u_x\partial_u, \\ [v_2]_Q &= -u_t\partial_u, \\ [v_3]_Q &= -(u + xu_x + 2tu_t)\partial_u, \\ [v_4]_Q &= -(tu_x - 1)\partial_u, \\ [v_5]_Q &= (x - tu - txu_x - t^2u_t)\partial_u. \end{aligned}$$

The prolongations of these evolutionary representatives acting on the PDE Δ_B are:

$$\begin{aligned} \operatorname{pr}[v_1]_Q[\Delta_B] &= -D_x[\Delta_B], \\ \operatorname{pr}[v_2]_Q[\Delta_B] &= -D_t[\Delta_B], \\ \operatorname{pr}[v_3]_Q[\Delta_B] &= -(3 + xD_x + 2tD_t)[\Delta_B], \\ \operatorname{pr}[v_4]_Q[\Delta_B] &= -tD_x[\Delta_B], \\ \operatorname{pr}[v_5]_Q[\Delta_B] &= -t(3 + xD_x + tD_t)[\Delta_B]. \end{aligned} \tag{7}$$

However if one were to use the method of [1, 34], the resulting actions would instead be:

$$\begin{aligned} \operatorname{pr} v_1[\Delta_B] &= 0, \\ \operatorname{pr} v_2[\Delta_B] &= 0, \\ \operatorname{pr} v_3[\Delta_B] &= 3\nu\Delta_B, \\ \operatorname{pr} v_4[\Delta_B] &= 0, \\ \operatorname{pr} v_5[\Delta_B] &= 3\nu t\Delta_B. \end{aligned} \tag{8}$$

These calculations show that using the evolutionary representatives of the symmetries results in useful training signals, unlike previous methods used for loss augmentation.

B. Darcy Flow Symmetries

In this appendix, we provide detailed derivations of the evolutionary representatives and their prolongations for the Darcy flow symmetries.

The Darcy flow equation is given for $x \in D \subset \mathbb{R}^2$:

$$\Delta_D \coloneqq \nabla \cdot (k(x)\nabla u(x)) + f(x) = 0,$$

Unlike the the Burgers case, the algebra for the Darcy Flow is no longer finite dimensional. The algebra of symmetries for the Darcy flow equation includes the following generators:

$$v_1^\infty = h^{[1]}(u)\partial_u - k h_u^{[1]}(u)\partial_k,$$
$$v_2^\infty = -h_y^{[2]}(x,y)\partial_x - h_x^{[2]}(x,y)\partial_y + 2f h_{xy}^{[2]}(x,y)\partial_f,$$

where $h^{[1]}$ is arbitrary and $\nabla^2 h^{[2]}(x,y) = 0$.

The evolutionary representatives of these generators are:

$$\begin{aligned} v_1^\infty &= h^{[1]}(u)\partial_u - k h_u^{[1]}(u)\partial_k, \\ v_2^\infty &= -h_y^{[2]}(x,y)\partial_x - h_x^{[2]}(x,y)\partial_y \\ &\quad + 2f h_{xy}^{[2]}(x,y)\partial_f, \end{aligned}$$

The prolongations of these evolutionary representatives acting on the PDE are:

$$\begin{aligned} &\mathrm{pr}\,[v_1^\infty]_Q\,[\Delta_D] = 0, \\ &\mathrm{pr}\,[v_2^\infty]_Q\,[\Delta_D] = D_x[h_y^{[2]}\Delta_D] + D_y[h_x^{[2]}\Delta_D]. \end{aligned}$$

Unfortunately, by the virtue of already being in its evolutionary form, the first vector does not result in a useful training signal.

Now, if we are to enforce this during training we need to somehow sample $h^{[2]}$, such that it solves the Laplace equation. For practical purposes, we consider a finite-dimensional sub-algebra with linear $h^{[2]}$, resulting in a two-dimensional Lie algebra:

$$\begin{aligned} &\mathrm{pr}\,\left[v_2^0\right]_Q\,[\Delta_D] = D_y[\Delta_D], \\ &\mathrm{pr}\,\left[v_2^1\right]_Q\,[\Delta_D] = D_x[\Delta_D]. \end{aligned} \tag{9}$$

However if one were to use the method of [1, 34], the resulting actions would instead be:

$$\begin{aligned} &\mathrm{pr}\,v_1^\infty\,[\Delta_D] = 0, \\ &\mathrm{pr}\,v_2^\infty\,[\Delta_D] = 2h_{xy}^{[2]}\Delta_D, \end{aligned} \tag{10}$$

which for the two dimensional sub-algebra would both be zero, resulting in no training signal. Thus, one would be forced to solve the Laplace equation, making the resulting method significantly more complex.