# Group Averaging for Physics Applications: Accuracy Improvements at Zero Training Cost

**Valentino F. Foit**
Center for Cosmology and Particle Physics
Department of Physics
New York University
New York, NY 10003
foit@nyu.edu

**David W. Hogg**
Center for Cosmology and Particle Physics
Department of Physics
New York University
New York, NY 10003
david.hogg@nyu.edu

**Soledad Villar**
Department of Applied Mathematics & Statistics
and Mathematical Institute for Data Science
Johns Hopkins University
svillar3@jhu.edu

## Abstract

Many machine learning tasks in the natural sciences are precisely equivariant to particular symmetries. Nonetheless, equivariant methods are often not employed, perhaps because training is perceived to be challenging, or the symmetry is expected to be learned, or equivariant implementations are seen as hard to build. Group averaging is an available technique for these situations. It happens at test time; it can make any trained model precisely equivariant at a (often small) cost proportional to the size of the group; it places no requirements on model structure or training. It is known that, under mild conditions, the group-averaged model will have a provably better prediction accuracy than the original model. Here we show that an inexpensive group averaging can improve accuracy in practice. We take well-established benchmark machine learning models of differential equations in which certain symmetries ought to be obeyed. At evaluation time, we average the models over a small group of symmetries. Our experiments show that this procedure always decreases the average evaluation loss, with improvements of up to 37% in terms of the VRMSE. The averaging produces visually better predictions for continuous dynamics. This short paper shows that, under certain common circumstances, there are no disadvantages to imposing exact symmetries; the ML4PS community should consider group averaging as a cheap and simple way to improve model accuracy.

## 1 Introduction

One of the key contributions of machine learning to scientific communities has been emulators and surrogate models. These are data-driven approximations, typically deep neural networks, that learn the input-output mapping of a traditional numerical differential equation solver or physics-based simulation. Once trained, a surrogate model can predict the system's evolution at a fraction of the computational cost. Our work investigates how these surrogate models can be improved by enforcing some of the exact symmetries inherent in the underlying physical processes or equations.

It has been a long-standing and powerful idea in machine learning that models engineered to respect the symmetries of a problem should perform better. This concept has been formalized and given a rigorous theoretical foundation [1, 2, 3, 4]; that work proves a generalization benefit for a class of

models that are equivariant to a group action. In other words, a model that respects the symmetry will (under conditions) have a strictly lower test error than a model that does not.

To demonstrate this idea, we use the data and machine learning models presented in "The Well" [5], which is a collection of datasets containing numerical simulations of a wide variety of spatiotemporal physical systems. The project also includes trained state-of-the-art surrogate machine learning models. In this paper, we apply a simple and fast group averaging procedure to these trained models to produce new, exactly equivariant models [6, 7]. We show that this procedure improves the accuracy of models without any additional training. This both demonstrates the value of equivariance and also delivers equivariance (after the fact) at almost no cost. Here we use group averaging to make a very simple point, but in practice one should use the frame averaging technique, which implements symmetries by averaging over a small subset of group elements [8].

**Our contributions**  This paper is essentially a *demonstration* of the principle of group averaging, which transforms a generic machine learning model into a precisely equivariant model. Our demonstration shows that group averaging is practical and does improve model accuracy in a real physics context.

We emphasize that group averaging overcomes all of the standard objections to the use of equivariant approaches in machine learning: It places no constraints on model architecture or structure, and it places no burden on training. The discrete groups we use here — the dihedral groups — are small. For continuous groups we approximated the average by sampling only a small number of elements. So group averaging is very inexpensive.

## 2   Methods

### 2.1   Surrogate models

Surrogate models estimate the state variables of a spatiotemporal dynamical system, such as a parameterized family of PDEs with fixed parameters or the observations of a physical system. The solution to a system of $s$ state variables over a spatial domain $\mathbb{D}$ and for all times $t \geq 0$ is represented as $\mathbf{u} : \mathbb{D} \times [0, \infty) \to \mathbb{R}^s$.

The domain $\mathbb{D}$ is typically a subset of two- or three-dimensional space, such as $[0, L_x] \times [0, L_y]$ or $[0, L_x] \times [0, L_y] \times [0, L_z]$. We employ a standard uniform discretization of the system in time and space and denote by $N_D$ the number of elements in the discretized domain $\mathbb{D}$. A snapshot $\mathbf{u}_t : \mathbb{R} \to \mathbb{R}^{s \times N_D}$ represents the values of all $s$ state variables at all $N_D$ points in the discretized domain at a fixed time $t$.

The surrogate model $M : \mathbb{R}^{k \times s \times N_D} \to \mathbb{R}^{s \times N_D}$ is an autoregressive prediction problem that maps a sequence of $k$ snapshots $\mathbf{U}_{t_i} = [\mathbf{u}_{t_{i-k+1}}, \ldots, \mathbf{u}_{t_{i-1}}, \mathbf{u}_{t_i}]$ to the snapshot of the following time step such that

$$\mathbf{u}_{t_{i+1}} \approx \hat{\mathbf{u}}_{t_{i+1}} = M(\mathbf{U}_{t_i}). \tag{1}$$

The goal of the model $M$ is to minimize some distance between its predictions $\hat{\mathbf{u}}_t$ and the truth $\mathbf{u}_t$. The initial conditions at $t_0$ are given by $k$ snapshots $\mathbf{U}_{t_0}$, so that the estimated next sequence is updated according to $\hat{\mathbf{U}}_{t_1} = [\mathbf{u}_{t_{-k+2}}, \ldots, \mathbf{u}_{t_0}, \hat{\mathbf{u}}_{t_1}]$. The model $M$ is applied repeatedly to generate estimates for all desired times.

### 2.2   Equivariance from group averaging

Our goal is to improve already-trained models that do not inherently respect some particular symmetry of the problem, by group averaging. Let $G$ be a group. A function $f : X \to Y$ is $G$-equivariant if

$$f(\phi(g)x) = \psi(g)f(x) \quad \forall x \in X, \forall g \in G, \tag{2}$$

where $\phi$ is a representation of $G$ on $X$ and $\psi$ is a representation of $G$ on $Y$. In words, applying a transformation $\phi(g)$ to the input $x$ followed by the function $f$ is the same as applying the transformation $\psi(g)$ to $f(x)$. Equivariance is very common in physics, for example, all physically meaningful functions are equivariant with respect to coordinate transformations [9].

Let $G$ be a compact group. The unique invariant measure $\lambda$ of $G$ is called the Haar measure. It is normalized $\lambda(G) = 1$, and invariant so that for every measurable subset $A \subset G$ and for any $g \in G$ it holds $\lambda(gA) = \lambda(Ag) = \lambda(A)$. The function $f$ can be projected onto the space of equivariant functions using the Reynolds operator defined as

$$(\mathcal{Q}f)(x) = \int_G \psi(g^{-1}) f(\phi(g)x) \, d\lambda(g). \tag{3}$$

This map essentially averages the function $f$ over the group transformations. In the case of finite groups, the integral is a sum over the group elements. In [1] it is proven that this averaging procedure reduces the error of the predictions. The assumptions to obtain provable improvements are:

- The underlying physical problem is $G$-invariant. This often holds because governing equations satisfy symmetries.
- The distribution of the training data is $G$-invariant.

Our goal is to predict the dynamics of a system with a known symmetry. We can improve a surrogate model $M$ as in (1) with initial sequence $\mathbf{V}_{t_0} \equiv \mathbf{U}_{t_0}$ by group averaging

$$\hat{\mathbf{v}}_{t_{i+1}} = \frac{1}{|G|} \sum_{g \in G} \psi(g^{-1}) M(\phi(g)\mathbf{V}_{t_i}). \tag{4}$$

Our approach approximates the continuous group average (3) using a Monte Carlo method with $n$ random samples from its elements. We provide numerical examples for surrogate models in the next Section. We argue that this simple procedure provides an efficient, zero-shot method that allows us to inject physical knowledge into a model after it has been trained.

## 3 Numerical experiments

We use the simulations and models published in "The Well" [5]. The project provides 15 TB of simulation data across 16 datasets, ranging from biological systems, fluid dynamics, acoustic scattering, as well as magneto-hydrodynamic simulations of extra-galactic fluids and supernova explosions. We chose to analyze models corresponding to datasets defined on a Cartesian 2D grid with symmetric initial and boundary conditions. For an overview of the groups used in this work, see Appendix B. The dataset properties and state variables are summarized in Appendix C.

The `active_matter` dataset [10] consists of simulations of a continuum theory describing the dynamics of rod-like active particles immersed in a Stokes fluid. The simulations were run at constant particle density and varying values for the parameters dimensionless active dipole strength and strength of particle alignment through steric interactions. The initial conditions of the quantities are plane-wave perturbations about the isotropic state. The `gray_scott_reaction_diffusion` dataset (from now on abbreviated as `gray_scott`) contains numerical solutions of the Gray-Scott equations. They are a set of coupled reaction-diffusion equations that describe two chemical species, $A$ and $B$, whose concentrations vary in space and time. Two parameters control the "feed" and "kill" rates in the reaction. The initial conditions are either a random Fourier series or random clusters of Gaussians. The boundary conditions are periodic for both datasets. The underlying physical problems are sampled on a square grid and have the symmetry of the torus $\mathbb{T}^2$. Toroidal symmetry allows all variables to translate periodically in both directions. Additionally, the problem is invariant under rotations and inversions, which is the discrete symmetry of the square $D_4$.

Additionally we analyze two datasets on a rectangular, non-square grid. The boundary conditions are periodic in one direction and Neumann in the other. The `rayleigh_benard` dataset [11] contains simulations of convection between a hot and cold fluid layer. Initial random Gaussian noise creates density variations which causes fluid motion. The underlying problem has the symmetry of the circle $S^1$ as well as inversions $D_2$. The `turbulent_radiative_layer_2D` dataset [12] contains simulations of cold dense gas moving relative to hot dilute gas. Once turbulence is induced due to the Kelvin Helmholtz instability, mixing of the two layers occurs. While the equations governing this process are invariant under translations and inversions, the initial conditions are not sampled from a distribution that is invariant under inversions. In all samples of the training set, the gas moves preferred in one direction. Consequently, the surrogate model does not respect this symmetry either.

Table 1: Average loss values for the datasets. The different starting positions in the trajectory and the used groups are given. The loss values are the averages of the $\mathcal{L}_{\text{VRMSE}}$ over all trajectories and state variables in the respective test sets. The loss is evaluated at the time steps indicated above each column. The rollout loss is the sum of the first 15 errors after the initial sequence. The continuous symmetries $S^1$ and $\mathbb{T}^2$ are approximated with $n = 1$ random group elements, except in (d), as indicated.

| Model | Start | 1 | 5 | 10 | Rollout |
|---|---|---|---|---|---|
| Baseline | | 0.063 | 0.386 | 0.938 | 10.81 |
| $D_4$ | 10 | **0.035** | **0.236** | **0.594** | **7.012** |
| $\mathbb{T}^2$ | | 0.066 | 0.319 | 0.803 | 9.277 |
| Baseline | | 0.038 | 0.279 | 0.663 | 7.659 |
| $D_4$ | 50 | **0.026** | **0.193** | **0.497** | **5.821** |
| $\mathbb{T}^2$ | | 0.038 | 0.202 | 0.507 | 5.976 |

(a) `active_matter`

| Model | Start | 1 | 10 | 50 | Rollout |
|---|---|---|---|---|---|
| Baseline | | 0.008 | 0.154 | 0.742 | 1.833 |
| $D_4$ | 10 | **0.007** | 0.146 | 0.680 | 1.730 |
| $\mathbb{T}^2$ | | 0.008 | **0.126** | **0.598** | **1.511** |
| Baseline | | 0.004 | 0.073 | 0.530 | 0.879 |
| $D_4$ | 50 | **0.003** | 0.064 | 0.453 | 0.764 |
| $\mathbb{T}^2$ | | 0.004 | **0.051** | **0.357** | **0.608** |

(b) `gray_scott`

| Model | Start | 1 | 10 | 20 | Rollout |
|---|---|---|---|---|---|
| Baseline | | 0.065 | 0.346 | 0.627 | 4.246 |
| $D_2$ | 50 | **0.054** | 0.311 | 0.589 | 3.795 |
| $S^1$ | | 0.065 | **0.294** | **0.526** | **3.664** |
| Baseline | | 0.035 | 0.243 | 0.436 | 2.908 |
| $D_2$ | 100 | **0.032** | 0.216 | 0.394 | 2.586 |
| $S^1$ | | 0.034 | **0.198** | **0.363** | **2.395** |

(c) `rayleigh_benard`

| Model | Start | 1 | 5 | 10 | Rollout |
|---|---|---|---|---|---|
| Baseline | | **0.142** | 0.350 | 0.579 | 7.834 |
| $S^1_{n=1}$ | 0 | 0.171 | 0.355 | 0.537 | 7.813 |
| $S^1_{n=8}$ | | 0.144 | **0.306** | **0.487** | **6.911** |
| Baseline | | 0.201 | 0.499 | 0.744 | 9.128 |
| $S^1_{n=1}$ | 50 | 0.204 | 0.485 | 0.730 | 8.939 |
| $S^1_{n=8}$ | | **0.174** | **0.449** | **0.722** | **8.605** |

(d) `turbulent_radiative_layer_2D`

The predictive power of the trained models vary greatly between the different datasets and models. Group averaging provides no improvement for the `shear_flow` dataset because the specific initial conditions break the system's underlying symmetry, even though the theory itself and the boundary conditions are symmetric.

The CNextU-Net [13, 14] was chosen in the analysis of the datasets due to its consistent performance. The predictions $\hat{\mathbf{u}}_t$ and $\hat{\mathbf{v}}_t$ generated by (1) and (4) must be compared to the simulations $\mathbf{u}_t$. To measure how well a prediction fits the simulation, we employ the variance scaled root mean squared error (VRMSE)

$$\mathcal{L}_{\text{VRMSE}}(\mathbf{u}, \mathbf{v}) = \left[ \frac{\langle |\mathbf{u} - \mathbf{v}|^2 \rangle}{\langle |\mathbf{v} - \langle \mathbf{v} \rangle|^2 \rangle + \epsilon} \right]^{1/2}, \tag{5}$$

where $\langle \cdot \rangle$ denotes the spatial mean operator, and $\epsilon = 10^{-7}$ to prevent division by zero in case $\text{Var}(\mathbf{v}) = 0$. Lower VRMSE indicates higher agreement. $\mathcal{L}_{\text{VRMSE}} \gtrsim 1$ signifies poor performance, as it indicates a prediction that is less accurate than a naively predicting the constant mean.

**Results**

We evaluate the performance improvement of the group averaging method by comparing the errors of the *baseline* prediction $\mathcal{L}_{\text{VRMSE}}(\hat{\mathbf{u}}_t, \mathbf{u}_t)$ and of the *equivariant* prediction $\mathcal{L}_{\text{VRMSE}}(\hat{\mathbf{v}}_t, \mathbf{u}_t)$. Figure 1 provides illustrations. The results are summarized in Table 1 and Figure 2.

We experimented with the number of random Monte Carlo samples, $n$, used to approximate the continuous groups. We observe significant improvements even at $n = 1$, demonstrating an enhancement over the baseline surrogate models at no additional computational cost. Table 1d shows how the group average improves as $n$ increases.

The group averaging procedure outperforms the baseline prediction on all time frames, even if we sum over only one random representative of a continuous group per time step. At sufficiently late times, the models lose predictive power and the loss approaches 1. The rate of loss growth and the optimal averaging group are model-dependent.

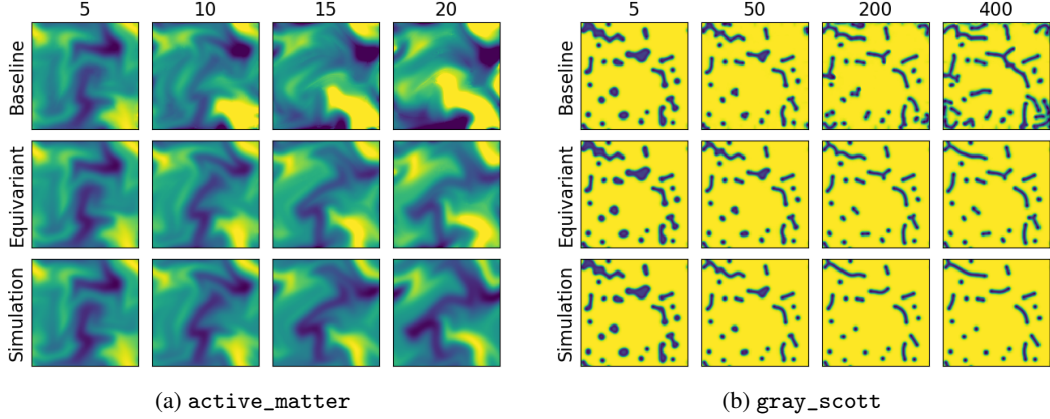(a) `active_matter`　　　　　　　　　　　　　　(b) `gray_scott`

Figure 1: Example comparisons of time evolutions of the concentration of two datasets. (a) After $t = 10$ steps we visually start to observe the benefits of group averaging. The equivariant model has a lower loss than the baseline at all times and is closer to the simulation (truth). At later times significant deviations between the approaches become apparent. (b) The equivariant model captures the features of the simulation relatively well for long times, while the baseline model develops several fictitious clusters.
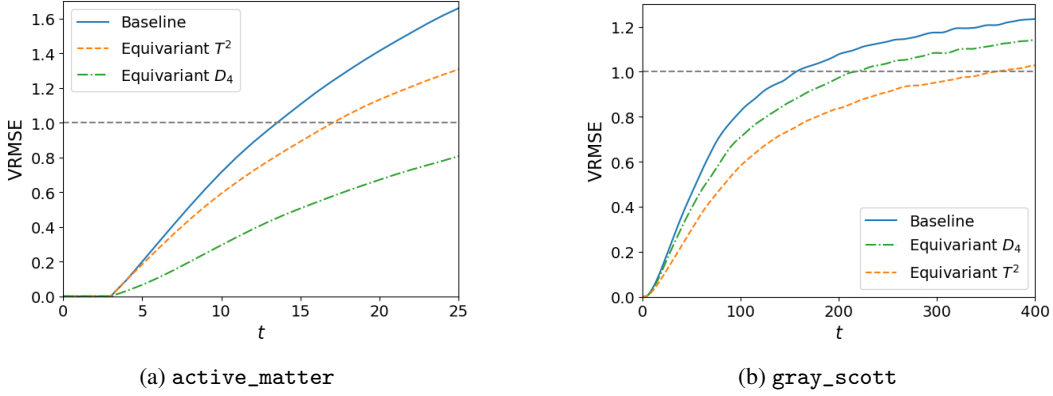


(a) `active_matter`　　　　　　　　　　　　　　(b) `gray_scott`

Figure 2: Losses for the concentration predictions. In both cases the translations of $\mathbb{T}^2$ are performed with a single ($n = 1$) random translation per time step.

## 4  Discussion

Imposing known symmetries theoretically improves the generalization error of machine learning models under mild conditions. However, in practice, equivariant models are not always better, maybe because it can be harder to train equivariant models than established non-equivariant models. Group averaging is a well-established alternative technique that delivers equivariance at no training cost and a post-processing cost that scales with the size of the group. Here we show that for certain physics applications it improves the models significantly. Thus, when symmetry groups are small, there is no reason to sacrifice accuracy with non-equivariant surrogate models and emulators.

The dihedral groups we consider here are very small. The continuous groups we consider are approximated by sampling only a small number of elements. The methods used here could be improved for continuous groups, for example, by taking a subgroup or a frame of the continuous group [8]. It remains to be seen if the average could be improved by using a Wasserstein barycenter, which is designed to preserve geometric structures in the data [15]. Finally, many important groups for physics are not compact at all, such as translations, units transformations, boosts, coordinate diffeomorphisms, and variable reparameterizations. Non-compact groups are out of scope for group averaging, though some averaging techniques can be extended to non-compact groups using the Weyl unitarian trick [16].

# References

[1] Bryn Elesedy and Sheheryar Zaidi. Provably strict generalisation benefit for equivariant models. In *International Conference on Machine Learning*, pages 2959–2969. PMLR, 2021.

[2] Mircea Petrache and Shubhendu Trivedi. Approximation-generalization trade-offs under (approximate) group equivariance. *Advances in Neural Information Processing Systems*, 36:61936–61959, 2023.

[3] Behrooz Tahmasebi and Stefanie Jegelka. The exact sample complexity gain from invariances for kernel regression. *Advances in Neural Information Processing Systems*, 36:55616–55646, 2023.

[4] Ningyuan Huang, Ron Levie, and Soledad Villar. Approximately equivariant graph networks. *Advances in Neural Information Processing Systems*, 36:34627–34660, 2023.

[5] Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina Agocs, Miguel Beneitez, Marsha Berger, Blakesly Burkhart, Stuart Dalziel, Drummond Fielding, et al. The well: A large-scale collection of diverse physics simulations for machine learning. *Advances in Neural Information Processing Systems*, 37:44989–45037, 2024.

[6] Dmitry Yarotsky. Universal approximations of invariant maps by neural networks. *Constructive Approximation*, 55(1):407–474, 2022.

[7] Ryan Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Relational pooling for graph representations. In *International Conference on Machine Learning*, pages 4663–4673. PMLR, 2019.

[8] Omri Puny, Matan Atzmon, Heli Ben-Hamu, Ishan Misra, Aditya Grover, Edward J Smith, and Yaron Lipman. Frame averaging for invariant and equivariant network design. In *10th International Conference on Learning Representations, ICLR 2022*, 2022.

[9] Soledad Villar, David W Hogg, Weichi Yao, George A Kevrekidis, and Bernhard Schölkopf. Towards fully covariant machine learning. *Transactions on Machine Learning Research*, 2023.

[10] Suryanarayana Maddu, Scott Weady, and Michael J Shelley. Learning fast, accurate, and stable closures of a kinetic theory of an active fluid. *Journal of Computational Physics*, 504:112869, 2024.

[11] Keaton J Burns, Geoffrey M Vasil, Jeffrey S Oishi, Daniel Lecoanet, and Benjamin P Brown. Dedalus: A flexible framework for numerical simulations with spectral methods. *Physical Review Research*, 2(2):023068, 2020.

[12] Drummond B Fielding, Eve C Ostriker, Greg L Bryan, and Adam S Jermyn. Multiphase gas and the fractal nature of radiative turbulent mixing layers. *The Astrophysical Journal Letters*, 894(2):L24, 2020.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[14] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[15] Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. In *International conference on machine learning*, pages 685–693. PMLR, 2014.

[16] Soledad Villar, Weichi Yao, David W Hogg, Ben Blum-Smith, and Bianca Dumitrascu. Dimensionless machine learning: Imposing exact units equivariance. *Journal of Machine Learning Research*, 24(109):1–32, 2023.

[17] Kimberly Stachenfeld, Drummond B Fielding, Dmitrii Kochkov, Miles Cranmer, Tobias Pfaff, Jonathan Godwin, Can Cui, Shirley Ho, Peter Battaglia, and Alvaro Sanchez-Gonzalez. Learned coarse models for efficient turbulence simulation. *arXiv preprint arXiv:2112.15275*, 2021.

[18] Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Holden Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, et al. Multiple physics pretraining for physical surrogate models. *arXiv preprint arXiv:2310.02994*, 2023.

[19] Liu Yang, Siting Liu, Tingwei Meng, and Stanley J Osher. In-context operator learning with data prompts for differential equation problems. *Proceedings of the National Academy of Sciences*, 120(39):e2310142120, 2023.

[20] Md Ashiqur Rahman, Robert Joseph George, Mogab Elleithy, Daniel Leibovici, Zongyi Li, Boris Bonev, Colin White, Julius Berner, Raymond A Yeh, Jean Kossaifi, et al. Pretraining codomain attention neural operators for solving multiphysics pdes. *Advances in Neural Information Processing Systems*, 37:104035–104064, 2024.

[21] Jingmin Sun, Yuxuan Liu, Zecheng Zhang, and Hayden Schaeffer. Towards a foundation model for partial differential equations: Multioperator learning and extrapolation. *Physical Review E*, 111(3):035304, 2025.

[22] Junhong Shen, Tanya Marwah, and Ameet Talwalkar. Ups: Efficiently building foundation models for pde solving via cross-modal adaptation. *arXiv preprint arXiv:2403.07187*, 2024.

[23] Maximilian Herde, Bogdan Raonic, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. *Advances in Neural Information Processing Systems*, 37:72525–72624, 2024.

[24] Tianyu Chen, Haoyi Zhou, Ying Li, Hao Wang, Chonghan Gao, Rongye Shi, Shanghang Zhang, and Jianxin Li. Omniarch: Building foundation model for scientific computing. *arXiv preprint arXiv:2402.16014*, 2024.

[25] Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. *Advances in Neural Information Processing Systems*, 35:1596–1611, 2022.

[26] Jeroen Audenaert, Micah Bowles, Benjamin M Boyd, David Chemaly, Brian Cherinka, Ioana Ciucă, Miles Cranmer, Aaron Do, Matthew Grayling, Erin E Hayes, et al. The multimodal universe: enabling large-scale machine learning with 100tb of astronomical scientific data. *arXiv preprint arXiv:2412.02527*, 2024.

[27] John Shawe-Taylor. Building symmetries into feedforward networks. In *1989 First IEE International Conference on Artificial Neural Networks,(Conf. Publ. No. 313)*, pages 158–162. IET, 1989.

[28] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pages 2990–2999. PMLR, 2016.

[29] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International conference on machine learning*, pages 2747–2755. PMLR, 2018.

[30] Daniel Worrall and Max Welling. Deep scale-spaces: Equivariance over scale. *Advances in Neural Information Processing Systems*, 32, 2019.

[31] Rui Wang, Robin Walters, and Rose Yu. Incorporating symmetry into deep dynamics models for improved generalization. *arXiv preprint arXiv:2002.03061*, 2020.

[32] Mario Geiger and Tess Smidt. e3nn: Euclidean neural networks. *arXiv preprint arXiv:2207.09453*, 2022.

[33] Wilson G Gregory, Kaze WK Wong, and Soledad Villar. Robust emulator for compressible navier-stokes using equivariant geometric convolutions. In *Machine Learning and the Physical Sciences Workshop*. NeurIPS, 2024.

[34] Ben Blum-Smith and Soledad Villar. Machine learning and invariant theory. *Notices of the American Mathematical Society*, 70(08):1–1, 2023.

[35] Soledad Villar, David W Hogg, Kate Storey-Fisher, Weichi Yao, and Ben Blum-Smith. Scalars are universal: Equivariant machine learning, structured like classical physics. *Advances in neural information processing systems*, 34:28848–28863, 2021.

[36] Ben Blum-Smith, Ningyuan Huang, Marco Cuturi, and Soledad Villar. A galois theorem for machine learning: Functions on symmetric matrices and point clouds via lightweight invariant features. *arXiv preprint arXiv:2405.08097*, 2024.

[37] Wilson G Gregory, Josué Tonelli-Cueto, Nicholas F Marshall, Andrew S Lee, and Soledad Villar. Learning equivariant tensor functions with applications to sparse vector recovery. *arXiv preprint arXiv:2406.01552*, 2024.

[38] Taco S Cohen, Mario Geiger, and Maurice Weiler. A general theory of equivariant cnns on homogeneous spaces. *Advances in Neural Information Processing Systems*, 32, 2019.

[39] Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions. In *International Conference on Machine Learning*, pages 15546–15566. PMLR, 2023.

[40] Taco S Cohen and Max Welling. Steerable cnns. *arXiv preprint arXiv:1612.08498*, 2016.

[41] Risi Kondor. N-body networks: a covariant hierarchical neural network architecture for learning atomic potentials. *arXiv preprint arXiv:1803.01588*, 2018.

## A   Related work

There has been a lot of recent work in machine learning for surrogate models, and they have demonstrated good performance in many domains [17, 18, 19, 20, 21, 22, 23, 24]. These successes have motivated the creation of diverse datasets, such as [25, 5, 18, 26].

There is a vast literature on symmetry-preserving machine learning models [27, 28, 29]. Some authors have incorporated symmetry directly into the network architecture to create equivariant networks of different types [30, 31, 4, 32, 33]. Several approaches to building equivariant models have been discussed, such as using invariant theory [34, 35, 36, 37], group convolutions [28, 38], canonicalization [39], and irreducible representations [40, 41]. In this work, we impose symmetries simply by averaging over the group orbit [6, 7]. This is a fairly naive approach that was significantly improved by recent work proposing *frame averaging*, which only requires averaging over a subset of the group [8].

Incorporating symmetries into machine learning models has been proven to improve generalization error and to reduce the sample complexity [1, 2, 3]. However, those are theoretical results that don't necessarily apply to complicated architectures such as those based on representation theory or group convolutions. In [1] it is mathematically shown that if the target function is equivariant, and the training data is sampled independently from an invariant distribution, then the projection (3) reduces the generalization error.

## B   Groups

We are interested in datasets on a square or rectangular Cartesian grid. Datasets defined on a square grid with appropriate initial and boundary conditions have the largest possible symmetry.

### B.1   Dihedral groups

The symmetry group of the square is $D_4$ and contains $|D_4| = 8$ elements. For datasets defined on a rectangle, the biggest symmetry group is $D_2$, containing $|D_2| = 4$ elements. Inversions correspond to $D_1 \cong C_2 \cong Z_2$ with $|D_1| = 2$ elements.

Generally, the dihedral groups $D_n$ refer to the symmetry groups of the $n$-gon, groups of order $2n$. We can think of these as $n$ rotational symmetries and $n$ reflection symmetries.

The group representation of $D_4$ is

$$D_4 = \langle r, i | r^4 = i^2 = (ir)^2 = e \rangle, \tag{6}$$

where its members can be thought of as rotations and inversions. One representation is given by the matrices

$$R = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \quad I = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{7}$$

and the $|D_4| = 8$ group elements can be constructed by

$$\mathbb{1}, R, R^2, R^3, I, IR, IR^2, IR^3. \tag{8}$$

$D_2$ can be constructed as above using only even powers of $R$, and $D_1$ by using only the inversion.

## B.2   Lie groups

Translation invariant problems with periodic boundary conditions obey the symmetry of the circle. The circle group is an example of a compact continuous Lie group. It is known by several names, such as $T \cong S^1 \cong U(1) \cong SO(2)$. The irreducible representations are one-dimensional maps to the unit circle. If the problem is translation invariant and periodic in two directions, it is defined on a 2-torus. The torus group $\mathbb{T}^2$ is the continuous Abelian group formed from the direct product of two circle groups $\mathbb{T}^2 \cong S^1 \times S^1$.

Since numerical simulation data is always given on a discrete grid of dimension $N \times M$, we approximate the continuous groups with discrete cyclic groups $C_N \cong \mathbb{Z}/\mathbb{Z}_N$, that is $S^1$ with $C_N$ and $\mathbb{T}^2$ with $C_N \times C_M$.

## B.3   A vector is something that transforms like a vector

We now look at how scalars, vectors, and tensors transform under the transformations of the aforementioned group elements. Since we are interested in datasets defined on a 2D Cartesian grid, 2-dimensional matrix representations of the group elements are useful.

Let us assume $\phi \equiv \phi(g)$ is a matrix representation of $g \in G$.

Scalars do not transform

$$c \to c' = c. \tag{9}$$

Vectors transform like

$$\mathbf{u} \to \mathbf{u}' = \phi \mathbf{u}. \tag{10}$$

Tensors transform like

$$\mathbf{D} \to \mathbf{D}' = \phi \mathbf{D} \phi^{\mathsf{T}}. \tag{11}$$

For example, the orientation tensor $\mathbf{D}$ transforms under a $\pi/2$ rotation using (7) as

$$\mathbf{D}' = R \mathbf{D} R^{\mathsf{T}} = \begin{pmatrix} D_{yy} & -D_{yx} \\ -D_{xy} & D_{xx} \end{pmatrix}. \tag{12}$$

Unlike rotations or reflections, translations do not alter or mix the components of a vector or tensor. The entire fields are simply shifted.

## C   Experimental details

"The Well"[1] [5] is published under CC BY 4.0. The baseline surrogate models were each trained on a specific dataset for a fixed duration of 12 hours on a single Nvidia H100 GPU.

Each test set contains $0.1 \times N$ of the trajectories. We excluded 9 trajectories from the `gray_scott` dataset that converge to a steady state in our analysis.

Table 2: Dataset specifications

| Dataset | Size (GB) | $N_D$ | $T$ | $N$ | $s$ |
|---|---|---|---|---|---|
| active_matter | 51.3 | $256 \times 256$ | 81 | 360 | 11 |
| gray_scott | 153.8 | $128 \times 128$ | 1001 | 1200 | 2 |
| rayleigh_benard | 342 | $512 \times 128$ | 200 | 1750 | 4 |
| turbulent_radiative_layer_2D | 6.9 | $384 \times 128$ | 101 | 90 | 4 |

(a) Total size, resolution $N_D$, number of time steps per trajectory $T$, number of trajectories in the data set $N$, number of state variables $s$.

| Dataset | Quantity | Type | Components |
|---|---|---|---|
| active_matter | Concentration | Scalar | 1 |
| | Velocity | Vector | 2 |
| | Orientation | Tensor | 4 |
| | Rate-of-strain | Tensor | 4 |
| gray_scott | Concentration $A$ | Scalar | 1 |
| | Concentration $B$ | Scalar | 1 |
| rayleigh_benard | Buoyancy | Scalar | 1 |
| | Pressure | Scalar | 1 |
| | Velocity | Vector | 2 |
| turbulent_radiative_layer_2D | Buoyancy | Scalar | 1 |
| | Pressure | Scalar | 1 |
| | Velocity | Vector | 2 |

(b) The state variables of the datasets.

---

[1] `https://polymathic-ai.org/the_well/`