Smooth and Sparse Latent Dynamics in Operator Learning with Jerk Regularization

Anonymous Author(s)
Affiliation
Address
email

Abstract

Data-driven latent dynamics models (LDMs) offer a promising approach for fast and accurate spatiotemporal forecasting by computing solutions in a compressed latent space. However, these models often neglect temporal correlations between consecutive snapshots when constructing the latent space, leading to suboptimal compression, jagged latent trajectories, and limited extrapolation ability over time. To address these issues, this paper introduces a continuous operator learning framework that incorporates a novel jerk regularization into the learning of the compressed latent space. This jerk regularization promotes smoothness and sparsity of latent space dynamics, which not only yields enhanced accuracy and convergence speed but also helps identify intrinsic latent space coordinates. The effectiveness of this framework is demonstrated through a two-dimensional unsteady flow problem governed by the Navier-Stokes equations, highlighting its potential to expedite high-fidelity simulations in various scientific and engineering applications.

1 Introduction

Modeling continuous spatiotemporal dynamics is critical across climate, fluids, biology, and engineering [1, 2, 3, 4]. While governing partial differential equations (PDEs) provide a principled foundation, deriving compact forms is often infeasible [5]. Data-driven surrogate models such as CNN/GNN-based models [6, 7, 8] and operator learning models such as DeepONet and FNO [9, 10, 11, 12, 13, 14] avoid reliance on closed-form PDEs, but both approaches typically rely on autoregressive time stepping, making long-horizon rollouts expensive [15]. In contrast, motivated by the fact that spatiotemporal systems often evolve on low-dimensional manifolds [16, 12], latent dynamics models (LDMs) aim to model the dynamics in a compressed latent space learned via an autoencoder, enabling cheaper forecasts than autoregressive operator methods [17, 18, 19, 20, 21, 22, 23, 24, 25, 26].

To enhance the robustness and reduce the reliance on large datasets of purely data-driven methods, a promising approach is to embed physical principles such as the PDE itself [27, 28, 29, 30, 20, 31] or general principles such as invariance [32, 33, 34], equivariance [35], symmetry [36, 37], and conservation laws [38]. Among these, *a nearly universal property of physical systems is the temporal smoothness of their dynamics*. Thus, phenomena that evolve smoothly over time in the original state space should mirror this smooth trajectory in the reduced latent space. Autoencoders, however, always process data snapshot by snapshot independently, neglecting the temporal correlations between sequential snapshots and overlooking the fact that consecutive snapshots in latent space are likely more similar to each other than to those further away in time or from different trajectories. Prior attempts at integrating smoothness into the learning process either do not explicitly enforce smoothness or prove overly restrictive, penalizing even smooth spiraling trajectories [22, 21, 39]. In this work, we propose a spatiotemporal continuous operator-learning framework that directly minimizes jerk (the time derivative of acceleration) in latent space during autoencoder training, explicitly promoting smooth trajectories and implicitly inducing sparsity without penalties. Through a 2D unsteady Navier-Stokes example, we show that our jerk regularization yields faster, more accurate convergence for both the autoencoder and neural-ODE latent dynamics, smoother latent paths, and sparser, more interpretable coordinates.

2 Smooth operator learning

We consider a spatiotemporal system described by a PDE of the general form

$$\partial_t u(\boldsymbol{x}, t) + \mathcal{N}[u(\boldsymbol{x}, t)] = 0, \quad \boldsymbol{x} \in \Omega, \quad t \in [0, T], \tag{1}$$

where $u(\boldsymbol{x}, t)$ represents the continuous state of the system in the spatial domain $\Omega \subseteq \mathbb{R}^d$ and temporal interval $[0, T]$, with $\mathcal{N}$ a potentially unknown nonlinear differential operator. The system is further subjected to initial and boundary conditions, denoted by $u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x})$ and $u(\boldsymbol{x}_b, t) = g(\boldsymbol{x}_b, t)$ for $\boldsymbol{x}_b \in \partial\Omega$. Our primary goal is to learn the underlying dynamics of (1) from a training dataset, without requiring knowledge of the operator $\mathcal{N}$. We assume that the training dataset consists of various solution trajectories of (1), where the continuous states are discretized in space using a sampling grid $\{\boldsymbol{x}_i\}_{i=0}^N$ and in time using a fixed time interval Δt.

Our proposed model, depicted in Figure 1, adopts LDM-based continuous operator learning, merging nonlinear dimensionality reduction with continuous spatiotemporal modeling across space and time. During the inference stage, a CNN-based encoder transforms the initial high-dimensional system state $\boldsymbol{u}(0) = \{u(\boldsymbol{x}_i, 0)\}_{i=0}^N$ from the ambient space of dimension $N + 1$ into an initial latent vector $\boldsymbol{z}(0)$ in a compressed latent space of dimension $d_z \ll N$. Then, the low-dimensional latent vector is propagated through a neural ODE to compute a future latent vector $\boldsymbol{z}(t)$ at an arbitrary future time t. Propagating the latent vector instead of the high-dimensional state significantly accelerates the inference because the latent dimension d_z is significantly smaller than the ambient space dimension $N + 1$. Subsequently, a conditional INR-based decoder maps the future latent vector $\boldsymbol{z}(t)$ back to the predicted continuous state $\hat{\boldsymbol{u}}(\boldsymbol{x}, t)$, enabling predictions at arbitrary spatial and temporal resolutions.

The training process is split into two stages. During stage I, illustrated in Figure 1 (b), we train an autoencoder, consisting of an encoder E_θ with parameters θ and a decoder D_ϕ with parameters ϕ, to learn an invertible nonlinear map between the ambient space and the compressed latent space. We employ a standard mean-square-error reconstruction loss L^t_{recon} to ensure that the continuous states recovered from the decoder are similar to the high-dimensional states given to the encoder. Crucially, *we also ensure that trajectories in latent space are smooth* by including a jerk regularization loss that penalizes the third-order time derivative of the latent vector at each time t, which encourages changes in acceleration along the latent trajectory to be small without restricting the magnitude and direction of the velocity and acceleration. Using a finite difference approximation of the acceleration, the jerk regularization loss L^t_{jerk} at time t is calculated as

$$L^t_{\text{jerk}} = \|\boldsymbol{z}(t + 3\Delta t) - 3\boldsymbol{z}(t + 2\Delta t) + 3\boldsymbol{z}(t + \Delta t) - \boldsymbol{z}(t)\|_2^2, \tag{2}$$

where $\boldsymbol{z}(t)$, $\boldsymbol{z}(t+\Delta t)$, $\boldsymbol{z}(t+2\Delta t)$, and $\boldsymbol{z}(t+3\Delta t)$ are the latent vectors obtained by passing through the encoder a sequence of high-dimensional states $\boldsymbol{u}(t)$, $\boldsymbol{u}(t + \Delta t)$, $\boldsymbol{u}(t + 2\Delta t)$, and $\boldsymbol{u}(t + 3\Delta t)$ belonging to the same solution trajectory, and $\|\cdot\|_2$ denotes the L_2 norm of the vector. The total loss at time t for the training of the autoencoder is $L = L^t_{\text{recon}} + \lambda L^t_{\text{jerk}}$, where the scalar weight λ controls the strength of the jerk regularization.

Once the autoencoder is trained, we convert ambient space snapshots across all trajectories from the training dataset into their latent space representations. The resulting dataset of latent trajectories then serves as the basis for training the neural ODE during the training stage II, which is shown in Figure 1 (c). The neural ODE $\partial_t \boldsymbol{z}(t) = h_\psi[\boldsymbol{z}(t)]$ with parameters ψ is trained to learn the temporal evolution of latent vectors using a mean square error loss on whole trajectory rollouts. Finally, our final trained model enables predictions at any point in space and time, similar to other recent methods using both conditional INRs and neural ODEs [18, 40, 22, 19].

3 Results

We demonstrate the effectiveness of the proposed framework on a 2D problem governed by the incompressible Navier-Stokes equations in vorticity form with an added volume force term, corresponding

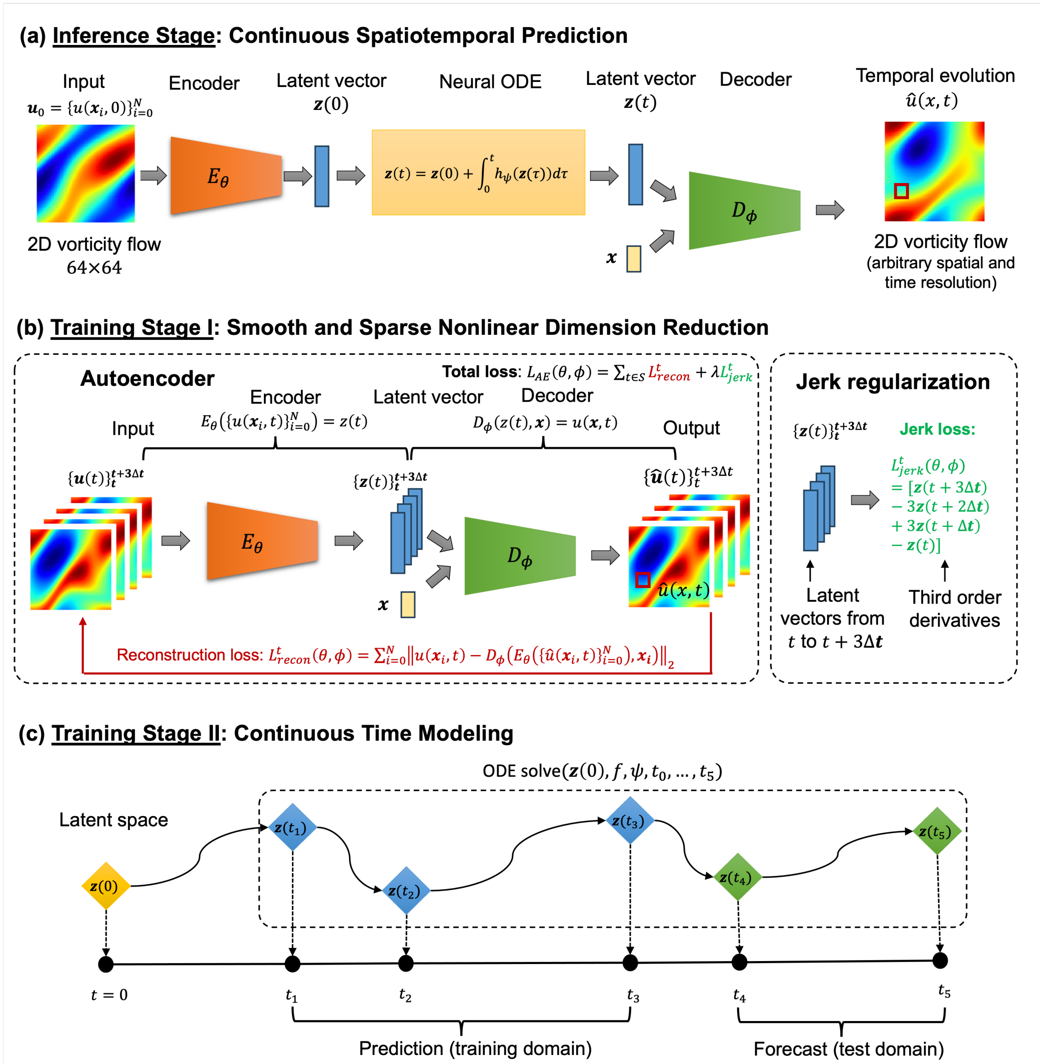

Figure 1: **Schematic of our smooth operator learning model. (a) Inference Stage**: The encoder transforms the initial system state $\boldsymbol{u}(0)$ into a compressed latent vector $\boldsymbol{z}(0)$, which is evolved through a neural ODE into a prediction $\boldsymbol{z}(t)$ at any subsequent time t. A conditional INR decoder then reconstructs the predicted continuous system state $u(\boldsymbol{x}, t)$ from $\boldsymbol{z}(t)$. **(c) Training Stage I**: The autoencoder learns a mapping between original and compressed latent spaces. A jerk regularization loss enforces smoothness of trajectories in latent space. **(c) Training Stage II**: The neural ODE is trained to learn a continuous-time model of the dynamics within the latent space.

to the $Re = 1000$ dataset investigated in the original FNO paper [9]. The original resolution of the simulation data is 256×256, which we downsampled to 64×64. Each trajectory contains 51 time steps with $\Delta t = 1$. We removed the initial 10 time steps due to inherent noise and diminished relevance to the underlying dynamics. The data from time $t = 11$ to $t = 40$ is designated for training, while data from $t = 41$ to $t = 50$ is set aside for time extrapolation testing. Of the 1000 total trajectories, 900 trajectories were used for training, and the remaining 100 trajectories for testing. We set the latent space dimension to $d_z = 32$.

Figure 2 displays the effect of jerk regularization on the trajectories in latent space. The left panel depicts a latent space trajectory in the test set obtained from an autoencoder trained after stage I without jerk regularization ($\lambda = 0$). The trajectory is visualized in terms of the time series of the components of the latent vector, which exhibit nonphysical jerkiness. The right panel shows that including jerk regularization ($\lambda = 0.1$) when training the autoencoder leads to a markedly smoother latent space trajectory, with a 35-fold drop in the average jerk metric, given by the average jerk loss over all time steps. Interestingly, jerk regularization also encourages sparsity of the latent vector, with most latent coordinates remaining constant over time. This suggests that jerk regularization may help selecting an optimal latent space dimension.

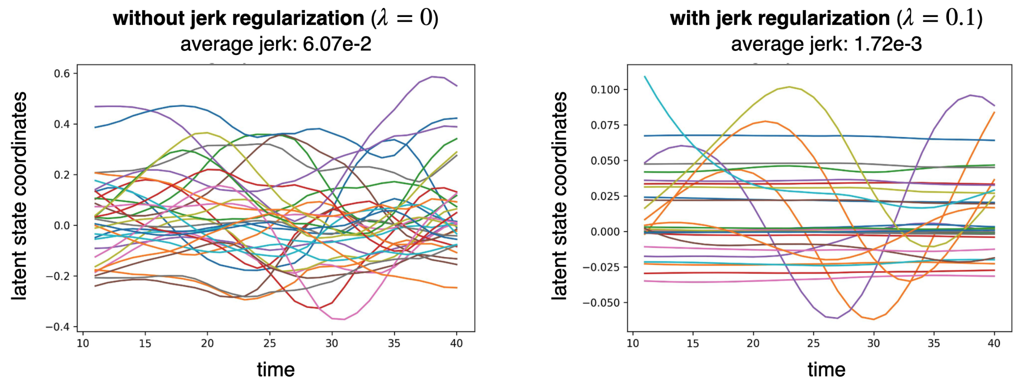

Figure 2: **Comparison of a latent space trajectory obtained with and without jerk regularization**. Jerk regularization yields smoother and sparser latent trajectories.

Figure 3 shows that the jerk regularization not only helps the autoencoder learn smoother latent space trajectories, but it yields increased accuracy of the reconstructed states.

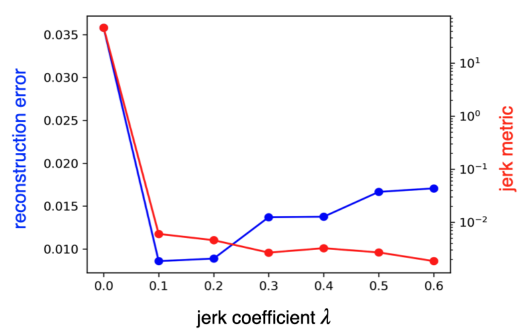

Figure 3: Reconstruction error and jerk metric for test trajectories versus jerk loss weight λ.

Finally, Figure 4 demonstrates that jerk regularization markedly improves the predictive accuracy of the final model obtained after training stages I and II, including beyond the training time domain. This improvement is due to the lower autoencoder reconstruction error as well as more accurate neural ODE training from smooth trajectories.

Overall, our smoothness-promoting jerk regularization shows potential to improve the performance of surrogate models for spatiotemporal dynamics. Future work will include evaluating the applicability of this technique across various physical systems.

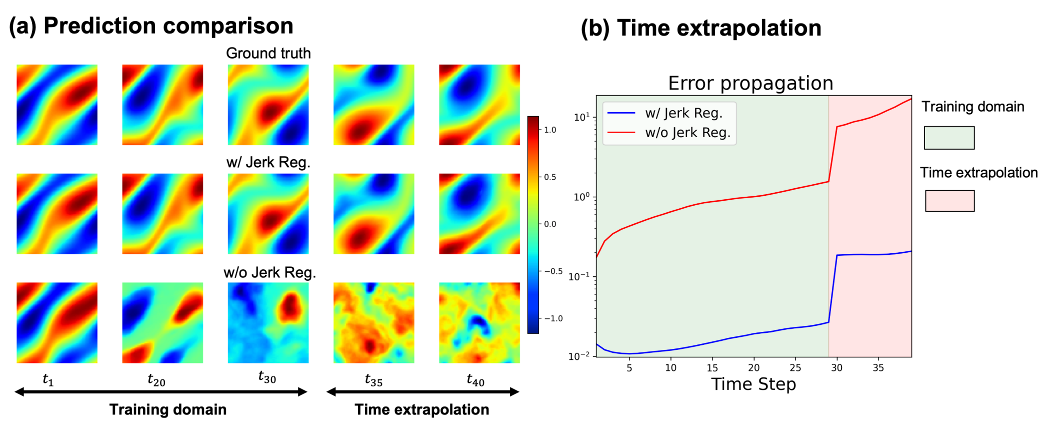

Figure 4: **Comparison of ground truth vorticity fields and model predictions**. **(a)** Time interpolation and extrapolation comparisons in the test set. **(b)** Relative mean square error (RMSE) inside (green area) and outside (pink area) the training time domain.

References

[1] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, 2023.

[2] Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual review of fluid mechanics*, 52:477–508, 2020.

[3] Mark Alber, Adrian Buganza Tepole, William R Cannon, Suvranu De, Salvador Dura-Bernal, Krishna Garikipati, George Karniadakis, William W Lytton, Paris Perdikaris, Linda Petzold, et al. Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. *NPJ digital medicine*, 2(1):115, 2019.

[4] Mojtaba Mozaffar, Shuheng Liao, Xiaoyu Xie, Sourav Saha, Chanwook Park, Jian Cao, Wing Kam Liu, and Zhengtao Gan. Mechanistic artificial intelligence (mechanistic-ai) for modeling, design, and control of advanced manufacturing processes: Current state and perspectives. *Journal of Materials Processing Technology*, 302:117485, 2022.

[5] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[6] Pu Ren, Chengping Rao, Yang Liu, Jian-Xun Wang, and Hao Sun. Phycrnet: Physics-informed convolutional-recurrent network for solving spatiotemporal pdes. *Computer Methods in Applied Mechanics and Engineering*, 389:114399, 2022.

[7] Chengping Rao, Pu Ren, Qi Wang, Oral Buyukozturk, Hao Sun, and Yang Liu. Encoding physics to learn reaction-diffusion processes. *Nature Machine Intelligence*, pages 1–15, 2023.

[8] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.

[9] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[10] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

[11] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

[12] Jacob Seidman, Georgios Kissas, Paris Perdikaris, and George J Pappas. Nomad: Nonlinear manifold decoders for operator learning. *Advances in Neural Information Processing Systems*, 35:5601–5613, 2022.

[13] Tapas Tripura and Souvik Chakraborty. Wavelet neural operator for solving parametric partial differential equations in computational mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 404:115783, 2023.

[14] Georgios Kissas, Jacob H Seidman, Leonardo Ferreira Guilhoto, Victor M Preciado, George J Pappas, and Paris Perdikaris. Learning operators with coupled attention. *The Journal of Machine Learning Research*, 23(1):9636–9698, 2022.

[15] Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022.

[16] Albert Cohen and Ronald DeVore. Approximation of high-dimensional parametric pdes. *Acta Numerica*, 24:1–159, 2015.

[17] Kookjin Lee and Kevin T Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, 2020.

[18] Peter Yichen Chen, Jinxu Xiang, Dong Heon Cho, Yue Chang, GA Pershing, Henrique Teles Maia, Maurizio M Chiaramonte, Kevin Thomas Carlberg, and Eitan Grinspun. Crom: Continuous reduced-order modeling of pdes using implicit neural representations. In *The Eleventh International Conference on Learning Representations*, 2022.

[19] Yuan Yin, Matthieu Kirchmeyer, Jean-Yves Franceschi, Alain Rakotomamonjy, and Patrick Gallinari. Continuous pde dynamics forecasting with implicit neural representations. In *The Eleventh International Conference on Learning Representations*, 2023.

[20] Aleksei Sholokhov, Yuying Liu, Hassan Mansour, and Saleh Nabi. Physics-informed neural ode (pinode): embedding physics into models using collocation points. *Scientific Reports*, 13(1):10166, 2023.

[21] Xiaolong He, Youngsoo Choi, William D Fries, Jonathan L Belof, and Jiun-Shyan Chen. Glasdi: Parametric physics-informed greedy latent space dynamics identification. *Journal of Computational Physics*, page 112267, 2023.

[22] Zhong Yi Wan, Leonardo Zepeda-Núñez, Anudhyan Boral, and Fei Sha. Evolve smoothly, fit consistently: Learning smooth latent dynamics for advection-dominated systems. *arXiv preprint arXiv:2301.10391*, 2023.

[23] Louis Serrano, Lise Le Boudec, Armand Kassaï Koupaï, Thomas X Wang, Yuan Yin, Jean-Noël Vittaut, and Patrick Gallinari. Operator learning with neural fields: Tackling pdes on general geometries. *arXiv preprint arXiv:2306.07266*, 2023.

[24] Ryan V Raut, Zachary P Rosenthal, Xiaodan Wang, Hanyang Miao, Zhanqi Zhang, Jin-Moo Lee, Marcus E Raichle, Adam Q Bauer, Steven L Brunton, Bingni W Brunton, et al. Arousal as a universal embedding for spatiotemporal brain dynamics. *bioRxiv*, pages 2023–11, 2023.

[25] Samuel E Otto, Gregory R Macchio, and Clarence W Rowley. Learning nonlinear projections for reduced-order modeling of dynamical systems using constrained autoencoders. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(11), 2023.

[26] Peiyi Chen, Tianchen Hu, and Johann Guilleminot. A nonlinear-manifold reduced-order model and operator learning for partial differential equations with sharp solution gradients. *Computer Methods in Applied Mechanics and Engineering*, 419:116684, 2024.

[27] Xiaoyu Zhao, Zhiqiang Gong, Yunyang Zhang, Wen Yao, and Xiaoqian Chen. Physics-informed convolutional neural networks for temperature field prediction of heat source layout without labeled data. *Engineering Applications of Artificial Intelligence*, 117:105516, 2023.

[28] 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. *arXiv preprint arXiv:2111.03794*, 2021.

[29] 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.

[30] Somdatta Goswami, Aniruddha Bora, Yue Yu, and George Em Karniadakis. Physics-informed deep neural operator networks. In *Machine Learning in Modeling and Simulation: Methods and Applications*, pages 219–254. Springer, 2023.

[31] N Navaneeth, Tapas Tripura, and Souvik Chakraborty. Physics informed wno. *Computer Methods in Applied Mechanics and Engineering*, 418:116546, 2024.

[32] Julia Ling, Reese Jones, and Jeremy Templeton. Machine learning strategies for systems with invariance properties. *Journal of Computational Physics*, 318:22–35, 2016.

[33] 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.

[34] Xiaoyu Xie, Arash Samaei, Jiachen Guo, Wing Kam Liu, and Zhengtao Gan. Data-driven discovery of dimensionless numbers and governing laws from scarce measurements. *Nature communications*, 13(1):7562, 2022.

[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] Alex Gabel, Victoria Klein, Riccardo Valperga, Jeroen SW Lamb, Kevin Webster, Rick Quax, and Efstratios Gavves. Learning lie group symmetry transformations with neural networks. In *Topological, Algebraic and Geometric Learning Workshops 2023*, pages 50–59. PMLR, 2023.

[37] Hee-Sun Choi, Yonggyun Yu, and Hogeon Seo. Symmetry-informed surrogates with data-free constraint for real-time acoustic wave propagation. *Applied Acoustics*, 214:109686, 2023.

[38] Derek Hansen, Danielle C Maddix, Shima Alizadeh, Gaurav Gupta, and Michael W Mahoney. Learning physical models that can respect conservation laws. *arXiv preprint arXiv:2302.11002*, 2023.

[39] Saaketh Desai, Ankit Shrivastava, Marta D'Elia, Habib N Najm, and Rémi Dingreville. Trade-offs in the latent representation of microstructure evolution. *Acta Materialia*, 263:119514, 2024.

[40] Shaowu Pan, Steven L Brunton, and J Nathan Kutz. Neural implicit flow: a mesh-agnostic dimensionality reduction paradigm of spatio-temporal data. *Journal of Machine Learning Research*, 24(41):1–60, 2023.