PROSE-FD: A Multimodal PDE Foundation Model for Learning Multiple Operators for Forecasting Fluid Dynamics

Yuxuan Liu
Department of Mathematics
University of California, Los Angeles
`yxliu@math.ucla.edu`

Jingmin Sun
Department of Mathematical Sciences
Carnegie Mellon University
`jingmins@andrew.cmu.edu`

Xinjie He
Department of Mathematics
University of California, Los Angeles
`xinjieh@math.ucla.edu`

Griffin Pinney
Department of Mathematics
University of California, Los Angeles
`griffinpinney@math.ucla.edu`

Zecheng Zhang
Department of Mathematics
Florida State University
`zecheng.zhang.math@gmail.com`

Hayden Schaeffer
Department of Mathematics
University of California, Los Angeles
`hayden@math.ucla.edu`

Abstract

We propose PROSE-FD, a zero-shot multimodal PDE foundational model for simultaneous prediction of heterogeneous two-dimensional physical systems related to distinct fluid dynamics settings. These systems include shallow water equations and the Navier-Stokes equations with incompressible and compressible flow, regular and complex geometries, and different buoyancy settings. This work presents a new transformer-based multi-operator learning approach that fuses symbolic information to perform operator-based data prediction, i.e. non-autoregressive. By incorporating multiple modalities in the inputs, the PDE foundation model builds in a pathway for including mathematical descriptions of the physical behavior. We pre-train our foundation model on 6 parametric families of equations collected from 13 datasets, including over 60K trajectories. Our model outperforms popular operator learning, computer vision, and multi-physics models, in benchmark forward prediction tasks. We test our architecture choices with ablation studies.

1 Introduction

Fluid dynamics and related physical models are essential for describing a wide range of scientific phenomena and are employed in various applications, including aerodynamics and aircraft design, weather forecasting, petroleum flow, space plasma dynamics and safety, combustion, and more. Simulating and predicting fluid dynamics is often complex and computationally expensive due to the system's highly nonlinear and multiscale behavior, such as the chaotic effects seen in turbulent flows. This challenge is amplified in real-world applications where measurements of state variables are scarce and noisy, some physical variables are unobserved, and boundary effects are unknown.

The code is available at: `https://github.com/felix-lyx/prose`.

Foundation Models for Science Workshop,38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Consequently, a key task in scientific machine learning is to develop models capable of learning general solution operators for fluid dynamics that can handle these issues.

Operator learning methods are a popular approach to train neural networks as surrogate models for solutions of partial differential equations (PDEs). These methods aim to train deep neural networks (DNNs) to approximate the map from input functions, such as boundary data and initial states, to the solution of the physical system. One advantage of neural operators is their potential for improved cost efficiency during inference [29, 40]. Recent advancements in operator learning include the Deep Operator Network (DeepONet) [24, 25, 20] and the Fourier Neural Operator (FNO) [18], which have demonstrated promise in scientific applications such as fluid dynamics [17] and weather prediction [29].

A key challenge with operator learning methods is that they are designed to train a single-operator network for one physical system at a time. Since training a deep neural network requires a large amount of data, this process often necessitates costly simulations or experiments. Moreover, because the operator network is trained on a single physical system, the resulting models do not exhibit emergent generalizations of physical properties [39]. PDE foundation models have emerged as a potential solution to address this by incorporating multiple physical systems into one model.

Foundation models in natural language processing and computer vision are deep learning models trained for multiple tasks using large datasets sampled from heterogeneous sources [3]. Approaches such as BERT [6], GPT [30, 31, 4], DALL-E [33, 32], Stable Diffusion [34], and LLAMA [44, 45], as well as Claude, have demonstrated success in data processing and generative tasks, showing evidence of generalization to new downstream tasks. However, these models have not been directly applicable to scientific computing problems, such as solving forward and inverse problems in PDEs, which require a higher degree of accuracy.

PDE foundation models aim to approximate solution operators for large classes of PDEs within a single DNN. The goal is to learn a general operator that can represent and infer the forward dynamics of distinct physical systems within one model. Thus, the fundamental task is to develop and train a DNN to accurately generalize to unseen physical dynamics that may share features with the trained dynamics. Current PDE foundation models include Predicting Operators and Symbolic Expressions (PROSE) [21, 39, 40], In-Context Operator Network (ICON) [50, 51, 52], Multiple Physics Pretraining (MPP) [27], Poseidon [11], Fourier Forecasting Network (FourCastNet) [29], and Aurora [2].

PROSE is a multimodal PDE foundation model that simultaneously learns to predict the values of state variables and derives a symbolic formulation of the governing equations describing the physical system [36, 37, 41]. PROSE has been applied to ordinary differential equations with chaotic behavior [21] and one-dimensional time-dependent nonlinear PDEs [39], and can incorporate robust fine-tuning and meta-learning strategies [40]. In [39, 40], it was shown that PROSE can generalize physical features to unseen conservation laws, though its capabilities in higher-dimensional PDE systems remain open. ICON employs in-context learning to guide the model in predicting state variables based on examples, which has been shown to generalize predictions for one-dimensional conservation laws. MPP pre-trains an autoregressive vision transformer [7, 8] to map observations to future states, although the formulation can be unstable for long prediction windows. For weather prediction, FourCastNet uses the Adaptive Fourier Neural Operator model [9], while Aurora utilizes the 3D Swin Transformer [22] to generate higher-resolution predictions. However, both models specialize in atmospheric forecasting, may lose some high-frequency information, and are tailored to specific problems. Another Swin-Transformer-based model is Poseidon [11], where the model needs to be fine-tuned for downstream tasks.

Main Contributions: We present *PROSE-FD*, a pre-trained PDE foundational model that uses a new transformer-based deep neural network that leverages state-variable observations and symbolic information to perform operator-based data prediction for fluid dynamics (FD). The model's formulation allows for the inclusion of various sources of information, including mathematical equations that describe the governing physics, in the inputs and/or outputs. Our contributions are listed below.

- We develop a multimodal fluids foundation model for predicting solution operators for shallow water equations and the Navier-Stokes equations with incompressible and compressible flow, regular and complex geometries, and different buoyancy settings.

- We show that the PROSE-FD model is capable of accurate predictions for fluid dynamics with a range of physical behavior.
- Using 13 datasets, we demonstrate that PROSE-FD is able to outperform single-operator learning approaches, computer vision models, and other multi-physics models, in the prediction and forecasting tasks with gains ranging from 1.3x to 7.9x.
- Our code and parameters are open-sourced for future experimentation and comparisons.

2 Methods

The main components of PROSE-FD include patch-based data encoding and decoding, symbolic equation encoding, and multimodal information fusion. We provide the problem description and the key components of PROSE-FD. We begin by summarizing related multimodal works.

2.1 Multimodal Machine Learning (MMML)

One key aspect of foundation models is that they are capable of multimodal machine learning (MMML) [23, 38, 43, 16, 49, 19], which focuses on building neural networks with the ability to comprehend, reason, and learn from diverse data sources and structures. As an example, for reasoning in image captioning generation, a multimodal model learns both visual features from the image and additional information from the corresponding textual data [47, 28]. While the success of foundation models in text-based tasks is well-established, their ability to reason and accurately represent quantities in scientific computing (SC) is limited. This is primarily due to the nature of the data, i.e. scientific data has lower information density than text, and due to the high precision requirements in SC problems. Specifically, there is a demand for a level of accuracy in scientific predictions that general-purpose foundation models are currently not designed to handle effectively. Some key challenges in MMML include (1) representation learning and (2) reasoning and generation. For representation learning, distinct but cooperative information from different modalities must be integrated into a uniform representation. PROSE-FD addresses this by utilizing a fusion layer with self-attention [46, 1, 48] sub-layers to enable information exchange between the two modalities for a holistic representation. For reasoning and generation, a model must provide a comprehensive understanding of the information gathered from different stages, such as representations from the fusion process and query locations. Cross-attention layers in the decoders facilitate this exchange and strengthen inter-modality relationships.

2.2 Problem Setting

Consider parametric families of two dimensional time-dependent nonlinear PDEs, whose state-variables of interest are represented by $\boldsymbol{u}(\boldsymbol{x}, t) \in \mathbb{R}^d$ (for some d up to 4 in our tests) with $\boldsymbol{x} \in \Omega \subseteq \mathbb{R}^2$. Given data up to T_0 timestamps, i.e. the sequence:

$$\{\boldsymbol{u}(\cdot, t_i) \mid 0 \le i < T_0\},$$

the goal of the forward problem is to predict the subsequent T timestamps

$$\{\boldsymbol{u}(\cdot, t_i) \mid T_0 \le i < T_0 + T\}.$$

More generally, for operator learning, the goal is to learn the solution as a map $(\boldsymbol{x}, t) \mapsto \boldsymbol{u}(\boldsymbol{x}, t)$ for $(\boldsymbol{x}, t) \in \Omega \times [T_0\Delta t, (T_0 + T)\Delta t]$ where $\Delta t = t_{i+1} - t_i$ is the timestep. In our experiments, we set $T_0 = T = 10$, that is, 10 timestamps are given as inputs and we predict, as operator evaluations, 10 future timestamps.

2.3 Model Overview

We use transformer layers as the backbone of the PROSE-FD model, similar to PROSE models introduced in [21, 39]. In Figure 1, we provide an illustration of our model, where the structure and the components in the gray box are similar to that of PROSE. The input data $\{\boldsymbol{u}(\cdot, t_i) \mid 0 \le i < T_0\}$ is first converted into patches before being mapped into a sequence of tokens, which is then processed with the Data Encoder consisting of transformer layers. The input equation symbols are tokenized into a sequence of word embeddings, which are then processed by the Symbol Encoder and fused

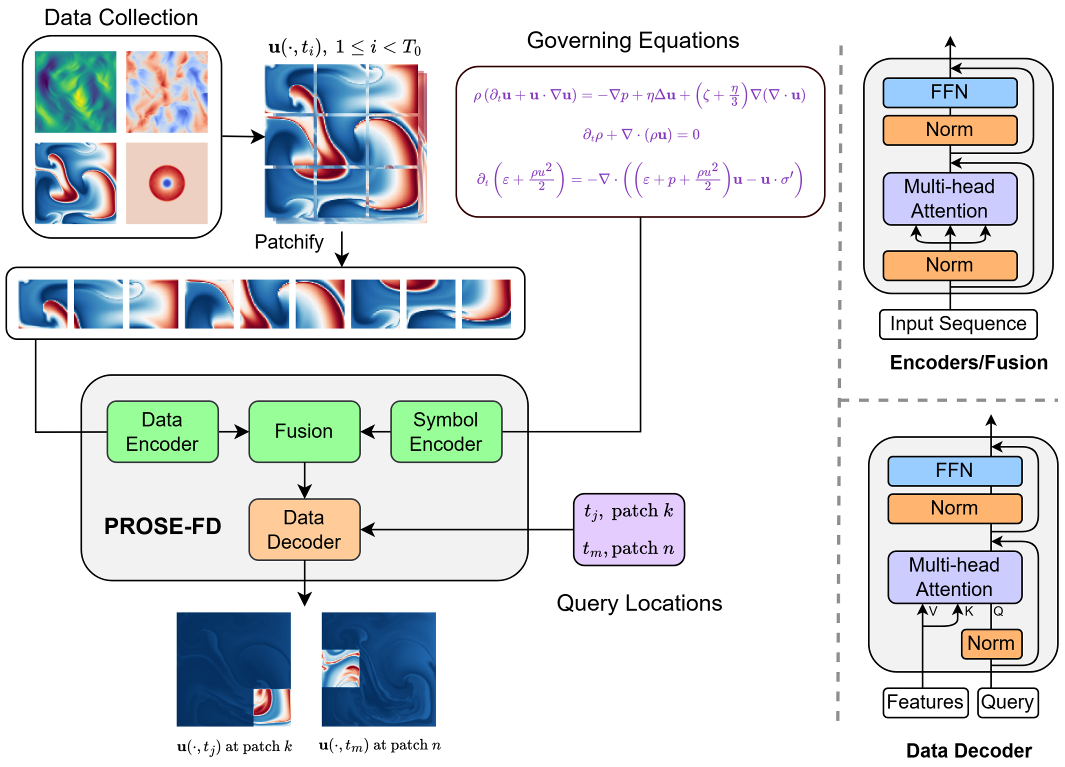

Data Collection
$\mathbf{u}(\cdot, t_i),\ 1 \le i < T_0$
Governing Equations
$\rho\left(\partial_t \mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \eta\Delta\mathbf{u} + \left(\zeta + \frac{\eta}{3}\right)\nabla(\nabla\cdot\mathbf{u})$
$\partial_t \rho + \nabla\cdot(\rho\mathbf{u}) = 0$
$\partial_t\left(\varepsilon + \frac{\rho u^2}{2}\right) = -\nabla\cdot\left(\left(\varepsilon + p + \frac{\rho u^2}{2}\right)\mathbf{u} - \mathbf{u}\cdot\sigma'\right)$
Patchify
Data Encoder
Fusion
Symbol Encoder
PROSE-FD
Data Decoder
t_j, patch k
t_m, patch n
Query Locations
$\mathbf{u}(\cdot, t_j)$ at patch k
$\mathbf{u}(\cdot, t_m)$ at patch n
FFN
Norm
Multi-head Attention
Norm
Input Sequence
Encoders/Fusion
FFN
Norm
Multi-head Attention
V
K
Q
Norm
Features
Query
Data Decoder

Figure 1: **PROSE-FD model overview.** The inputs to the model are the trajectories and governing equations (in symbolic form) sampled from the datasets. The input data is patchified before being converted into a sequence of features, which are then processed with encoders and fused with processed symbolic information. The data decoder takes in fused features and generates predictions at given query locations.

with data input in Fusion. The Data Decoder takes queries (based on t and the patch) along with the fused features to generate the solution at the specified locations. Notably, the model's computational complexity is linear with respect to the number of query locations, as each query is evaluated independently.

2.4 Patch-based Data Encoder

Processing the input data using transformers requires first converting the input into a sequence of tokens. For ODE systems or 1D PDE, directly projecting each timestamp into a token is an effective approach [21, 39]. However, for 2D PDE, a similar approach will lead to information loss due to the curse-of-dimensionality: for space resolution 128×128 and 3 channels (e.g. velocities and pressure), each token needs to encode information of dimension $128^2 \cdot 3 \approx 50K$, which is much larger than the hidden dimension. Patch-based encoding strategies were used in ViT [7] to balance the sequence length and token dimension. More precisely, each image (i.e. each timestamp $\boldsymbol{u}(\cdot, t)$) is first converted into p^2 patches each of resolution $(128/p) \times (128/p)$. These patches are then transformed and flattened into a sequence of p^2 tokens. We use $p = 8$, thus each token encodes spatial information of dimension $16^2 \cdot 3 = 768$, which is smaller than 1024, the hidden dimension of the linear and attention layers. Consequently, spatially dependent information is retained at the cost of a longer sequence.

We choose $p = 8$ for our inputs, thus the data input is first converted into a sequence of $64 \cdot T_0$ tokens. After adding learnable timestamps and patch positional encodings, the sequence is further processed using self-attention layers. Compared to other video transformer models such as Axial attention [12], this approach makes it easier to fuse information from different modalities later.

2.5 Equation Encoding and Fusion

An important aspect of the simultaneous learning of multi-operator solution operators is to encode the equations so that commonalities and differences can be automatically detected and processed. We encode the equations as symbolic trees [21, 39] (operations and functions as nodes, variables and constants as leaves), which are then converted to a sequence in Polish notation. The sequence is then processed similarly to sentences: they are converted to trainable tokens and then further processed using self-attention. The tokens retain meaning, i.e. mathematical functions such as "tan" and "add" are not further tokenized. Compared to LaTeX encoding of the equation, this approach has a shorter sequence length and easier syntax. For more details, we refer to [14, 5, 15].

To fuse information obtained from two input modalities, the processed data and symbol sequence are concatenated into a single sequence and further processed through self-attention in the Feature Fusion block. By attending to the symbols, the data sequence obtains information from the symbolic input (e.g., aspects of the equation underlying the data).

2.6 Patch-based Operator Decoder

For operator learning, the usual approach is to construct the solution as the map $(\boldsymbol{x}, t) \mapsto \boldsymbol{u}(\boldsymbol{x}, t)$. The PROSE-FD model constructs the solution via cross-attention, where the points $(\boldsymbol{x}, t)$ serve as queries, and the encoded features serve as keys and values. As a result of the curse-of-dimensionality issue, constructing the solution for all $(\boldsymbol{x}, t)$ with cross-attention is computationally expensive due to the increase in sequence length, i.e. the number of spatial points became the number of elements in the sequence. To ensure the independence of the query evaluation while maintaining reasonable computational complexity, instead of constructing a function evaluated for each input spacial location $\boldsymbol{x}$, the Data Decoder block learns a function that maps patches P. That is, given a patch P representing a set of spatial coordinates $P = \{\boldsymbol{x}_k\}_{k \in K}$ (where K is an index set with cardinality of 64 in our applications), the Data Decoder learn the solution $(P, t) \mapsto \{\boldsymbol{u}(\boldsymbol{x}_k, t_i) \mid \boldsymbol{x}_k \in P\}$. Additionally, due to the linear complexity of the Data Decoder, the output sequence length can be larger than the encoder's sequence length. Consequently, we use $p = 16$ output patches in each dimension.

3 Experiments

In this section, we first explain the experiment setup. We then present the main results and compare our PROSE-FD model with other baseline models. Finally, we validate our key architecture choices with ablation studies. More experiment details can be found in Appendix B.

3.1 Experiment Setup

Dataset. The dataset we use contains 6 parametric families of PDEs modeling fluid dynamics in different regimes collected from 3 heterogeneous sources: PDEBench [42], PDEArena [10], and CFDBench [26]. The dataset includes shallow water equations and the Navier-Stokes system with incompressible and compressible flow, regular and complex geometries, and different buoyancy settings. For datasets that do not provide a train/val/test splitting, we use the standard 80%/10%/10% splitting. For more details, we refer to Appendix A.

Evaluation Metric. The relative L^2 norm is used as the evaluation metric. More precisely, given the model's prediction $\tilde{\boldsymbol{u}}$ and the ground truth $\boldsymbol{u}$, we compute the (time-averaged) relative L^2 error:

$$\frac{1}{T} \sum_{i=T_0}^{T_0+T-1} \frac{\|\boldsymbol{u}(\cdot, t_i) - \tilde{\boldsymbol{u}}(\cdot, t_i)\|_2}{\|\boldsymbol{u}(\cdot, t_i)\|_2 + \varepsilon}, \tag{1}$$

where $T_0 = 10$ is the number of input steps, $T = 10$ is the number of output steps, and $\varepsilon = 10^{-7}$. Note that for the Navier-Stokes dataset from PDEArena, the temporal grid resolution is only 14, thus we set $T = 4$ for this dataset only. The average used in the last column of Table 1 is the average of the relative L^2 errors over the 6 families, i.e. the average over each row of the table.

Table 1: **Main Results and Comparisons with Baselines**. The numbers reported are relative L^2 errors (%). The averages are taken with respect to the 6 distinct families listed in the columns of the table. For each family of equations, we **bold** the best results. *Note that the PDEBench CNS contains 8 subsets of parameter configurations.

Model	Param	PDEBench			PDEArena		CFDBench	Average
		SWE	CNS*	INS	NS	NS-cond	-	
FNO	0.6M	3.71	6.31	36.83	38.68	55.63	8.53	24.95
DeepONet	3.5M	3.55	7.41	64.61	35.33	51.85	12.50	29.21
UNet	5.6M	0.33	3.19	3.43	12.56	16.82	0.76	6.18
ViT	154M	0.30	2.70	3.14	10.19	15.71	0.70	5.34
MPP-B	116M	1.02	1.90	7.52	5.71	12.56	1.23	4.99
PROSE-FD	165M	**0.28**	**1.41**	**2.75**	**5.27**	**9.61**	**0.61**	**3.32**

3.2 Baselines and Comparisons

We compare our PROSE-FD model with the following baselines. DeepONet [24] and FNO [18] are popular single-operator learning methods that efficiently approximate PDE solution operators. UNet [35] is a classical convolution-based image processing model, utilizing symmetric hierarchical structures to capture both context and fine details for pixel-wise predictions. ViT [7] is a popular transformer-based image processing model that captures global image dependencies and demonstrates scalability for model sizes in image and video processing tasks. MPP [27] is an Axial-ViT-based multi-physics pretraining approach, which autoregressively predicts PDE solutions. More details about the baselines are included in Appendix B.3.

3.3 Main Results

The main experiment results are included in Table 1, where we report the relative L^2 error (%) for each family of equations and the average. For all the models, we use the same training setting: a single model is trained to predict all families of equations, without any fine-tuning. Our PROSE-FD model exhibits remarkable performance, outperforming all baselines in all families of equations. We include example visualizations of the PROSE-FD model output in Figure 2 and Appendix B.4.

3.4 Ablation Studies

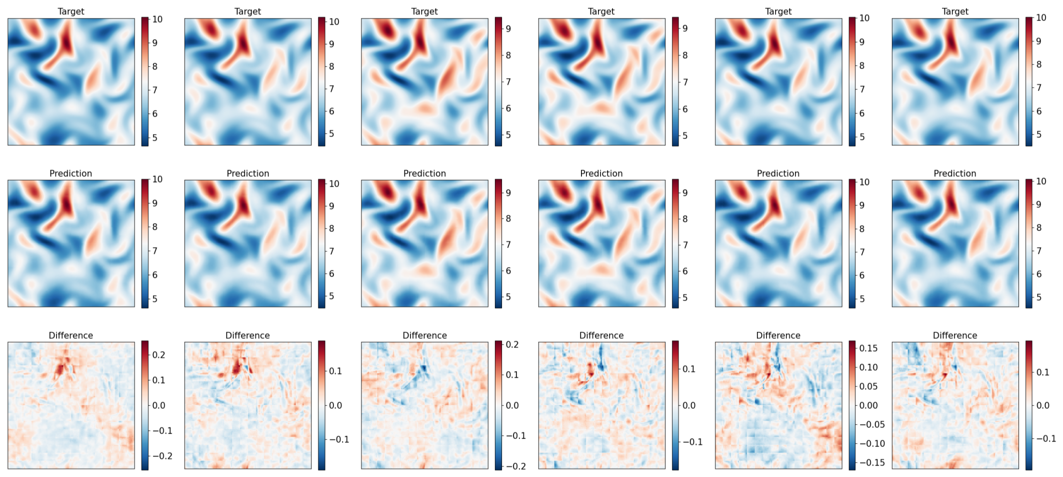
In this section, we present the results of our ablation studies to validate some key architecture choices. We compare the zero-shot testing performance (relative L^2 error) averaged over the 6 datasets. The results are shown in Table 2.

Rollout vs. Operator Formulation. Our baseline PROSE-FD model learns the map from T_0 input timestamps to T output steps in a non-autoregressive way, i.e., it can output multiple future timestamps in a single forward pass. The Data Decoder is an operator in time and no explicit rollout is needed. An alternative strategy is to have the model learn the map from T_0 input timestamps to 1 output step, then rollout, i.e. recursively apply, the model during the inference stage [8]. To obtain T output steps, the model needs to be called T times to generate the full solution. As shown in Table 2, our baseline model outperforms the rollout model, demonstrating that the operator learning approaches can avoid error accumulation.

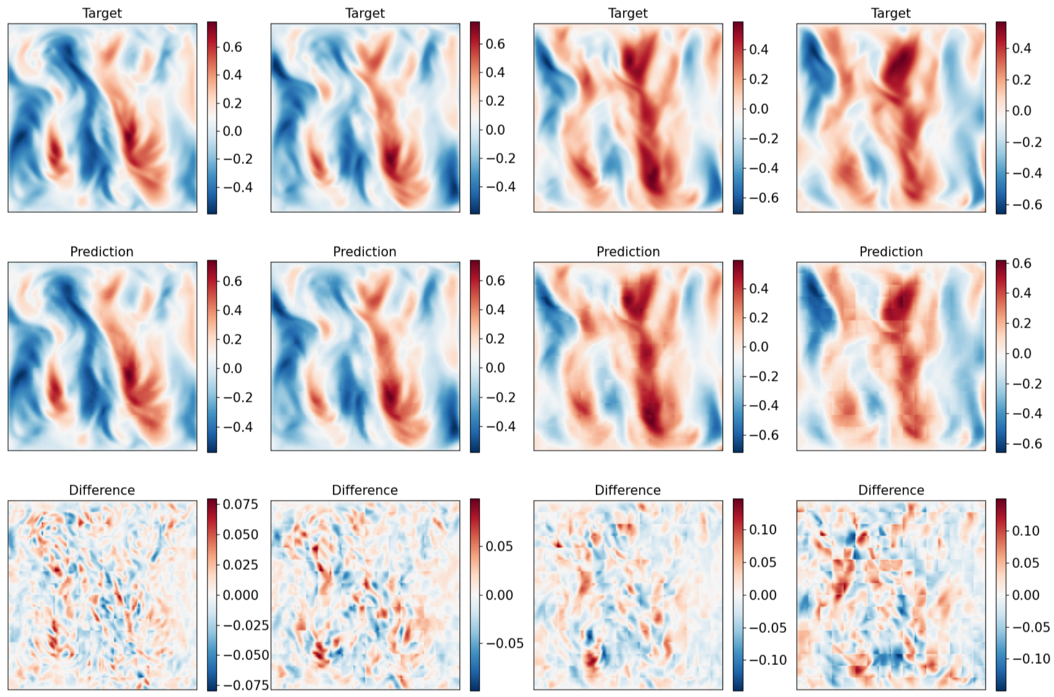
Influence of Symbolic Information. To show the importance of symbolic equation information, we compare our baseline PROSE-FD model with a model that only uses the Data Encoder and the Data Decoder. We increase the number of layers for each component so the final parameter count is close to the baseline model. The comparison in Table 2 demonstrates that the symbolic encoding structure of the PROSE-FD model enhances the predictions.

4 Conclusion

PROSE-FD is a pre-trained PDE foundational model that utilizes transformers to encode and process data and symbolic information for predicting solutions for fluid systems. The multimodal aspect of



(a) 5 consecutive output steps for PDEBench Compressible Navier-Stokes dataset. The channel plotted is the density field in equation (7). Each column represents a different timestamp. For this trajectory, the relative L^2 error is 0.34%.



(b) 4 consecutive output steps for PDEArena Navier-Stokes dataset. The channel plotted is the x-velocity in equation (10), i.e. u_x. Each column represents a different timestamp. For this trajectory, the relative L^2 error is 5.13%. This example shows that even when the relative error is higher, the structures of the flow can still be predicted correctly.

Figure 2: **Two example outputs for the PROSE-FD model.**

Table 2: **Results for PROSE-FD Ablation Studies.** We compare the baseline (operator) approach to a rollout (recursive) prediction variant and a data-only (no symbolic information) variant. The numbers reported are relative L^2 errors (%). The averages are taken with respect to the 6 distinct families listed in the columns of the table.

Model	Param	PDEBench			PDEArena		CFDBench	Average
		SWE	CNS	INS	NS	NS-cond	-	
Baseline	165M	0.28	1.41	2.75	5.27	9.61	0.61	3.32
Rollout in time	165M	0.23	1.16	4.56	4.48	8.86	1.08	3.39
Data-only	164M	0.29	1.54	2.49	6.43	10.39	0.54	3.61

the approach allows for further experimentation and enhancement by including additional modality information that describes the physical systems of interest. Our model is able to encode information from the two-dimensional shallow water equations and the two-dimensional Navier-Stokes equations with incompressible and compressible flow, regular and complex geometries, and different buoyancy settings into one model. Through extensive testing, we demonstrated that the model's predictions accurately capture the behavior presented in the datasets used – outperforming other single-operator learning and transformer models. Thus, the approach presents a general-purpose surrogate model for two-dimensional fluid systems. Further work will examine the scalability of the model and the encoding of boundary effects.

Acknowledgments

Y. Liu, X. He, G. Pinney, and H. Schaeffer were supported in part by NSF 2331033, NSF 2427558, and AFOSR MURI FA9550-21-1-0084. Z. Zhang was supported by DE-SC0025440.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Cristian Bodnar, Wessel P Bruinsma, Ana Lucic, Megan Stanley, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan Weyn, Haiyu Dong, Anna Vaughan, et al. Aurora: A foundation model of the atmosphere. *arXiv preprint arXiv:2405.13063*, 2024.

[3] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] Stéphane d'Ascoli, Pierre-Alexandre Kamienny, Guillaume Lample, and Francois Charton. Deep symbolic regression for recurrent sequences. *arXiv preprint arXiv:2201.04600*, 2022.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[8] Yalchin Efendiev, Wing Tat Leung, Guang Lin, and Zecheng Zhang. Efficient hybrid explicit-implicit learning for multiscale problems. *Journal of Computational Physics*, 467:111326, 2022.

[9] John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive fourier neural operators: Efficient token mixers for transformers. *arXiv preprint arXiv:2111.13587*, 2021.

[10] Jayesh K Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized pde modeling. *arXiv preprint arXiv:2209.15616*, 2022.

[11] Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. *arXiv preprint arXiv:2405.19101*, 2024.

[12] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

[13] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

[14] Zhongyi Jiang, Chunmei Wang, and Haizhao Yang. Finite expression methods for discovering physical laws from data. *arXiv preprint arXiv:2305.08342*, 2023.

[15] Pierre-Alexandre Kamienny, Stéphane d'Ascoli, Guillaume Lample, and Francois Charton. End-to-end symbolic regression with transformers. *Advances in Neural Information Processing Systems*, 35:10269–10281, 2022.

[16] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13401–13412, 2021.

[17] Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *arXiv preprint arXiv:2207.05209*, 2022.

[18] 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.

[19] Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. Foundations and trends in multimodal machine learning: Principles, challenges, and open questions. *arXiv preprint arXiv:2209.03430*, 2022.

[20] Guang Lin, Christian Moya, and Zecheng Zhang. B-deeponet: An enhanced bayesian deeponet for solving noisy parametric pdes using accelerated replica exchange sgld. *Journal of Computational Physics*, 473:111713, 2023.

[21] Yuxuan Liu, Zecheng Zhang, and Hayden Schaeffer. Prose: Predicting multiple operators and symbolic expressions using multimodal transformers. *Neural Networks*, 180:106707, 2024.

[22] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[23] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

[24] Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

[25] 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.

[26] Yining Luo, Yingfa Chen, and Zhen Zhang. Cfdbench: A comprehensive benchmark for machine learning methods in fluid dynamics. *arXiv preprint arXiv:2310.05963*, 2023.

[27] 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.

[28] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 299–307, 2017.

[29] 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.

[30] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[31] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[32] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[33] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021.

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[36] Hayden Schaeffer. Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2197):20160446, 2017.

[37] Hayden Schaeffer and Scott G McCalla. Sparse model selection via integral terms. *Physical Review E*, 96(2):023302, 2017.

[38] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7464–7473, 2019.

[39] Jingmin Sun, Yuxuan Liu, Zecheng Zhang, and Hayden Schaeffer. Towards a foundation model for partial differential equations: Multi-operator learning and extrapolation. *arXiv preprint arXiv:2404.12355*, 2024.

[40] Jingmin Sun, Zecheng Zhang, and Hayden Schaeffer. Lemon: Learning to learn multi-operator networks. *arXiv preprint arXiv:2408.16168*, 2024.

[41] Yifan Sun, Linan Zhang, and Hayden Schaeffer. Neupde: Neural network based ordinary and partial differential equations for modeling time-dependent data. In *Mathematical and Scientific Machine Learning*, pages 352–372. PMLR, 2020.

[42] 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.

[43] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[44] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[45] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[47] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France, 07–09 Jul 2015. PMLR.

[48] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[49] Peng Xu, Xiatian Zhu, and David A Clifton. Multimodal learning with transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[50] 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.

[51] Liu Yang, Tingwei Meng, Siting Liu, and Stanley J Osher. Prompting in-context operator learning with sensor data, equations, and natural language. *arXiv preprint arXiv:2308.05061*, 2023.

[52] Liu Yang and Stanley J Osher. Pde generalization of in-context operator networks: A study on 1d scalar nonlinear conservation laws. *arXiv preprint arXiv:2401.07364*, 2024.

A Dataset Details

The data was obtained from the PDEBench [42], PDEArena [10], and CFDBench [26] datasets. Unless otherwise specified, the space resolution is 128×128.

A.1 PDEBench [42]

Shallow Water Equation. The quantity of interest is the water depth $h(\boldsymbol{x}, t)$ on domain $[-2.5, 2.5]^2 \times [0, 1]$ with Neumann boundary condition. The temporal resolution is 101. The equations are:

$$\partial_t h + \nabla h\boldsymbol{u} = 0, \tag{2}$$

$$\partial_t h\boldsymbol{u} + \nabla \left(h\,\boldsymbol{u} \cdot \boldsymbol{u} + \frac{1}{2} g_r h^2 \right) = -g_r h \nabla b. \tag{3}$$

Incompressible Navier-Stokes Equation. The quantities of interest are the velocities $\boldsymbol{u}(\boldsymbol{x}, t)$ and particle density $c(\boldsymbol{x}, t)$ on domain $[0, 1]^2 \times [0, 5]$ with Dirichlet boundary condition. The temporal resolution is 1000. The equations are:

$$\rho(\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}) = -\nabla p + \mu \Delta \boldsymbol{u} + \mathbf{F}, \tag{4}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{5}$$

$$\partial_t c + \nabla \cdot (c\boldsymbol{u}) = 0. \tag{6}$$

The forcing term **F** is randomly sampled.

Compressible Navier-Stokes Equation. The quantities of interest are the velocities $\boldsymbol{u}(\boldsymbol{x}, t)$, pressure $p(\boldsymbol{x}, t)$, and density $\rho(\boldsymbol{x}, t)$ on domain $[0, 1]^2 \times [0, 1]$ with periodic boundary conditions. The temporal resolution is 21. For equations with low viscosities, the dataset is provided on a finer 512×512 space grid, which is downsampled to 128×128 for consistency (through average pooling). The equations are:

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{u}) = 0, \tag{7}$$

$$\rho(\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}) = -\nabla p + \eta \Delta \boldsymbol{u} + (\zeta + \eta/3) \nabla (\nabla \cdot \boldsymbol{u}), \tag{8}$$

$$\partial_t \left(\varepsilon + \frac{\rho u^2}{2} \right) = -\nabla \cdot \left(\left(\varepsilon + p + \frac{\rho u^2}{2} \right) \boldsymbol{u} - \boldsymbol{u} \cdot \sigma' \right). \tag{9}$$

A.2 PDEArena [10]

Incompressible Navier-Stokes Equation. The quantities of interest are the velocities $\boldsymbol{u}(\boldsymbol{x}, t)$ and particle density $c(\boldsymbol{x}, t)$ on domain $[0, 32]^2 \times [18, 102]$ with Dirichlet boundary conditions for velocity and Neumann boundary condition for particle field. The temporal resolution is 14. The equations are:

$$\rho(\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}) = -\nabla p + \mu \Delta \boldsymbol{u} + \mathbf{F}, \tag{10}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{11}$$

$$\partial_t c + \nabla \cdot (c\boldsymbol{u}) = 0. \tag{12}$$

The forcing term **F** takes the form $\mathbf{F} = (0, f)$ with $f = 0.5$.

Incompressible Navier-Stokes Equation (Conditioned). The quantities of interest are the velocities $\boldsymbol{u}(\boldsymbol{x}, t)$ and particle density $c(\boldsymbol{x}, t)$ on domain $[0, 32]^2 \times [18, 102]$ with Dirichlet boundary conditions for velocity and Neumann boundary condition for particle field. The temporal resolution is 56. The equations are:

$$\rho(\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}) = -\nabla p + \mu \Delta \boldsymbol{u} + \mathbf{F}, \tag{13}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{14}$$

$$\partial_t c + \nabla \cdot (c\boldsymbol{u}) = 0. \tag{15}$$

The forcing term **F** takes the form $\mathbf{F} = (0, f)$ where f is uniformly sampled in $[0.2, 0.5]$.

Table 3: **Model hyperparameters.** FFN means feedforward network.

Hidden dimension - attention	1024	Hidden dimension - FFN	2048
Number of attention heads	8	Fusion attention layers	8
Data encoder attention layers	2	Data decoder attention layers	8
Symbol encoder attention layers	4	Dropout	0
Input patch number	8	Output patch number	16
PreNorm	RMS Norm		

A.3 CFDBench [26]

Incompressible Navier-Stokes Equation. The quantities of interest are the velocities $\boldsymbol{u}(\boldsymbol{x}, t)$ and pressure $p(\boldsymbol{x}, t)$. This dataset contains irregular geometries with Dirichlet boundary conditions. The raw space resolution is 64×64 which is upsampled to 128×128 via interpolation. The equations are:

$$\rho(\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}) = -\nabla p + \mu \Delta \boldsymbol{u}, \tag{16}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{17}$$

B Experiment Details

We provide more details about the training process, architecture, and baselines.

B.1 Training

We perform data normalization during the training process. Given the input sequence of data $\{\boldsymbol{u}(\cdot, t_i) \mid 0 \leq i < T_0\}$, we compute the mean and standard deviation of each input trajectory, which are used to normalize both the input and ground truth sequence. The loss function is the standard mean squared error in the normalized space. The models are trained using the AdamW optimizer with a global batch size of 160 for 40 epochs where each epoch is 4,000 steps. The warmup-stable-decay learning rate scheduler [13] is used with 10% warmup and 20% decay. We use learning rate 10^{-4} and weight decay 10^{-4}. On two NVIDIA H100 GPUs, the training takes about 52 hours.

B.2 Model Hyperparameters

The model hyperparameters are summarized in Table 3.

B.3 Baselines

In this section, we include more details about the compared models.

DeepONet [24]. We employ the unstacked DeepONet architecture, consisting of a single trunk network and a single branch network. Initially, the input data is divided into 8×8 patches, with each patch being embedded into a 128-dimensional vector. These vectors are then passed through the branch network, producing an output with a basis dimension of $p = 50$. Simultaneously, the query point is processed through the trunk network, which also outputs a vector with the same dimension, p. The output solution at the query point is obtained by taking the inner product of the outputs from the two networks.

FNO [18]. We use 4 layers of standard 3d FNO to process the input data. The number of modes to keep in each dimension is set to 8, and the number of hidden channels is set to 16. The 3d FNO model directly maps 10 input steps to 10 output steps.

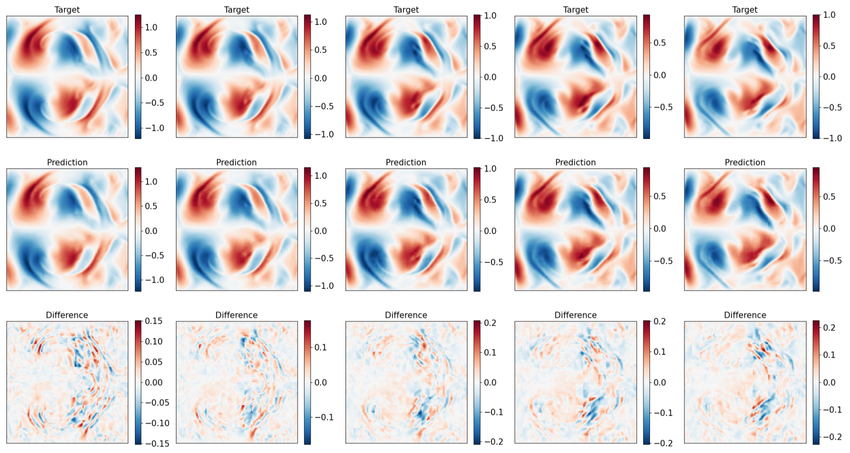
UNet [35]. We use 8 layers of 3d UNet with GeLU activation and 32 hidden dimensions. The 3d UNet model directly maps 10 input steps to 10 output steps (Padding is added to the input to make it a power of 2, which is needed for the shrinkage in dimension).

ViT [7]. For ViT, we use 10 layers of transformer encoder. The input patch number is set to be 8, the hidden dimension for attention is 1024, the hidden dimension for the feedforward network is 2048, and the number of heads is 8. The ViT model directly maps 10 input steps to 10 output steps.

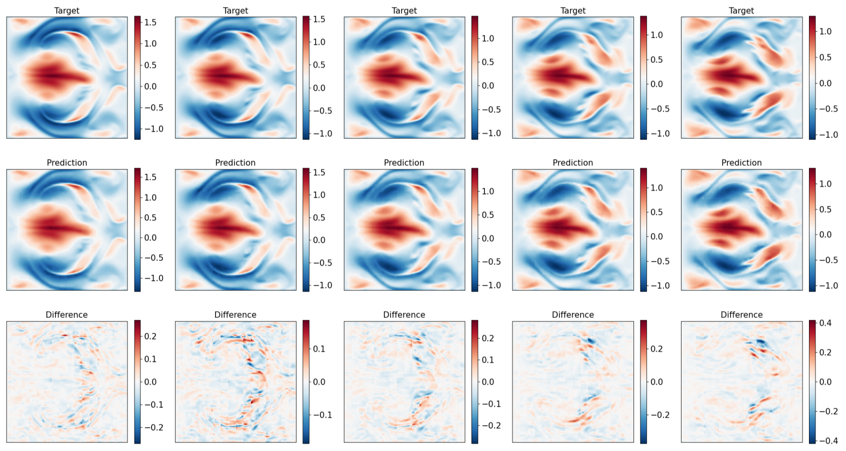
MPP [27]. In [27], MPP is trained on a different training dataset and different metrics are used. For a fair comparison, we retrained MPP-B using our training dataset using the same model configurations and optimizer hyperparameters. We evaluate MPP-B using the same testing setup and metric, where rollout is used to obtain all 10 output steps.

B.4 More Visualizations

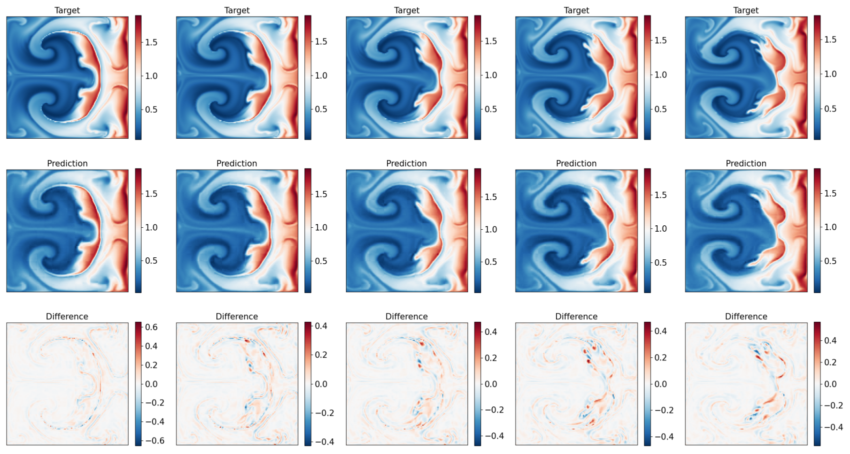
See Figure 3 for additional PROSE-FD model output visualizations.



(a) The channel plotted is the x-velocity.



(b) The channel plotted is the y-velocity.



(c) The channel plotted is the particle density.

Figure 3: **Example outputs for the PROSE-FD model.** 5 consecutive output steps for PDEArena NS-cond dataset (all three channels in equation (13)). Each column represents a different timestep. For this trajectory, the relative L^2 error is 8.74%.