Conditional Implicit Neural Representation for Operator Learning

Ruthwik Chivukula
School of Engineering, Brown University
Providence, RI, 02912, USA
CSCI 2470 Course Project
ruthwik_chivukula@brown.edu

Abstract

Operator Learning refers to learning a functional mapping between the input and output spaces of a Partial Differential Equation (PDE) in an infinite-dimensional setting. Neural Operators (NOs) approximate this mapping using neural network based backbones. Current state-of-the-art NO frameworks primarily rely on CNN or Transformer-based architectures, exploiting their spatial and temporal inductive biases, respectively to predict the solution maps of PDEs, which are generally spatio-temporal in nature. However, these methods introduce discretization and can be prohibitively expensive, both computationally and in terms of memory, when extended to 3D simulations involving complex phenomena such as turbulence. Moreover, many existing approaches suffer from the well-known spectral bias problem, wherein NOs tend to learn low frequency components more effectively while neglecting high frequency information that is crucial for modeling complex physical processes.

In this work, we explore the use of Implicit Neural Representations (INRs) for operator learning as a way to address these challenges. INRs are inherently resolution-agnostic due to their point-wise query-based formulation. We aim to leverage the advances in neural fields and neural representations from the vision community to mitigate key difficulties faced in scientific simulations using AI. The code implementation for this project can be found here: https://github.com/RC-circuit/CINR-for-Operator-Learning.

1. Introduction & Related Work

Operator learning aims to learn mappings between infinite-dimensional function spaces, typically arising from Partial Differential Equations (PDEs). Formally, one considers an operator

$$\mathcal{G} : \mathcal{A} \mapsto \mathcal{U}, \quad \mathcal{G}(a_i) = u_i, \quad a_i \in \mathcal{A},\, u_i \in \mathcal{U} \tag{1}$$

where a is an input function such as an initial condition, coefficient field, or forcing term, and u is the corresponding solution of the PDE defined over a spatial or spatio-temporal domain. Neural Operator (NO) models attempt to approximate $\mathcal{G}$ directly without relying on classical solvers. The learning objective is often expressed as an MSE between the predicted and ground-truth solution values queried at arbitrary points $\{a_i, u_i\}_{i=1}^{N}$, where $u_i = \mathcal{G}(a_i)$.

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \|\mathcal{G}(a_i) - \mathcal{G}_\theta(a_i)\|_2^2 \tag{2}$$

$\mathcal{G}_\theta$ is the learned operator parameterized by θ. This pointwise formulation highlights that operator learning does not intrinsically depend on any particular grid ordering.

A large body of recent operator-learning approaches build upon convolutional or attention-based architectures. Fourier Neural Operators (FNOs) [1] leverages FFT-based integral kernel parameterizations to capture long-range interactions efficiently. Convolutional Neural Operators (CNOs) [2] use convolution blocks but with a modification enabling generalization across different resolutions. Operator Transformers (OFormer) [3] introduce attention-based inductive biases to handle arbitrary query points unlike FNOs and CNOs which inherently are confined to structured grids unless additional non-trivial post-processing is done. However, it is known that attention-based models tend to overfit immediately in scarce data regimes, a common constraint in scientific machine learning, where the ground truths are produced using sophisticated compute-intensive numerical solvers. Hence, all of these methods, in one way or another, prevent scaling due to rapid growth of computational and memory costs, making high-resolution 2D and especially 3D simulations (e.g., turbulent flows) extremely challenging.

Another well-documented issue is the spectral bias of neural networks [4]. Many operator-learning models tend to predominantly learn low-frequency components of the solution manifold while struggling to capture the high frequency modes that are essential for accurately representing complex physics. This frequency imbalance becomes even more pronounced for turbulent or multi-scale problems.

To address these challenges, we aim to explore the scope and possibilities of adopting Implicit Neural Representations (INRs) for Operator Learning. INRs represent signals as continuous functions parameterized by neural networks and are queried at arbitrary spatial or spatio-temporal coordinates. This query-based formulation naturally makes them resolution-agnostic. The SIREN model [5], one of the initial works in INRs, demonstrated that periodic activations dramatically improve a network's ability to represent high frequency signals. Subsequent works introduced modulation strategies such as FiLM-inspired conditioning [6], enabling more controllable and expressive neural fields. The introduction of conditioning also allows these learned neural representations to generalize over a space of implicit functions.

We are not the first once to investigate INRs for scientific machine learning setting. INRs have been explored in this context for sometime now. Early neural field models like DeepSDF [7] established the auto-decoding paradigm, where each instance is associated with a learned latent code optimized jointly with the network weights. More recent works such as CORAL [8] have shown promising results in PDE modeling, leveraging the continuous nature of INRs to bypass strict grid requirements and mitigate discretization bottlenecks.

In this work, we explore how INR-based formulations can be extended further, specifically for the Operator Learning problems. We do a careful analysis of modulated conditional INRs and leverage some of the recent developments in the vision community in an attempt to push the boundaries of current state-of-the-art work. Our goal is to combine the resolution-agnostic representation power of INRs with the flexibility of neural fields and auto-decoding strategies to better capture complex, high frequency physical phenomena with fewer computational and memory bottlenecks.

2. Data

The PDE of interest for us here is the incompressible Navier Stokes equation. The initial condition for every trajectory is generated from a Gaussian random field. The external forcing term follows a sinusoidal form, $f(x,y) = 0.1\left(\sin\left(2\pi\left(x+y\right)\right)+\cos\left(2\pi\left(x+y\right)\right)\right)$.

The governing equations, continuity and momentum equations written in the vorticity form are as follows:

$$\nabla \cdot \mathbf{u} = 0 \tag{3}$$

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = \nu \Delta \omega + f \tag{4}$$

where $\mathbf{u}$ is the velocity field, ω is the scalar vorticity, ν is the viscosity parameter, and f is the external forcing term.

A total of 1600 samples are generated by rolling out forty different initial conditions, for forty time steps each with grid resolution: 128×128. The solution field that we aim to learn, $\omega(x,y,t)$ is a function of x, y, t, hence this is a 3D problem, 2D in space and 1D in time. Thus, we have $\omega_t \in \mathbb{R}^{128\times128}$. The data was generated using a pseudo-spectral semi-implicit Crank Nicolson numerical solver.

In traditional supervised learning for dynamical systems, the goal is to predict the future state ω_{t+1} given the current state ω_t. This can be written as a map, $\mathcal{G}(\omega_t) = \omega_{t+1}$ and the model is trained with a one step autoregressive strategy with teacher forcing. In contrast, the present work attempts a different formulation. Instead of directly predicting ω_{t+1} from ω_t, we aim to learn a latent representation z_t of the physical state and to propagate this representation in the latent space using a suitable sequential model. The latent state can then be decoded back to the physical field whenever needed.

The main focus of this project is to build a resolution-agnostic framework that projects a physical state ω_t into a latent state z_t through an encoder and reconstructs the physical state through a decoder with minimal reconstruction error. We wish to devise a pipeline that preserves the higher frequency modes during reconstruction without falling prey to spectral bias, which majority of the encoder-decoder methods fail to do. This approach can be scaled efficiently because the time integration takes place in the latent space, which is computationally inexpensive when compared to operating in the physical domain. Time integration in the latent space is left as a future extension. Preliminary attempts were made in this direction but were not explored fully within the scope of this project.

The learning setup is therefore self supervised. The input state ω_t is also the target output and the encoder and decoder are trained together to minimize the error between the reconstructed state and the original field. The dataset of 1600 samples is divided into training, validation and test sets with a split of $70\,\%$, $15\,\%$ and $15\,\%$ respectively.

3. Methodology

Our formulation is based on the ideas presented in [6]. The framework consists of three components, namely an encoder, a modulator and a synthesizer. These components are trained jointly to minimize the reconstruction error of the vorticity field. We discuss three different ways of posing the reconstruction problem, namely: Auto-encoder, Auto-decoder and lastly, a hybrid approach that uses both the encoder and auto-decoding, which hasn't been done before (as illustrated in Figure 1). We find that the hybrid approach outperforms the other two methods and plausible hypothesis for our finding is presented in the subsequent sub-section.

3.1. Components

3.1.1 Encoder

An explicit encoder is employed to extract a latent representation from the input field. In our work, the encoder is implemented as a CNN block. The architectural details follow the reference implementation provided in the codebase. We replace the ReLU activation with the GELU activation throughout the encoder, which empirically improves the reconstruction quality. The dimension of the latent vector produced by the encoder is treated as a hyper parameter, and an extensive ablation study is conducted in the following section to determine the optimal latent dimension.

Let ω_t denote the vorticity field at time t. The encoder produces an explicit latent embedding

$$z_t = E_\theta(\omega_t) \tag{5}$$

where θ represents the encoder parameters.

3.1.2 Modulator

The latent embedding is processed by a modulator network that implements a ReLU based multilayer perceptron. The role of the modulator is to generate scale and shift parameters that modulate the intermediate activations of the synthesizer. Specifically, for every hidden layer of the synthesizer, the modulator predicts two vectors, denoted by α and γ, which control amplitude scaling and phase shifting of the periodic activations. This is often referred to as FiLM (Feature-wise Linear Modulation) [9]. Since it produces both scale and shift values, the width of the modulator is chosen to be twice that of the corresponding synthesizer layer.

We denote the output of the modulator by

$$\alpha_t, \gamma_t = M_\phi(z_t) \tag{6}$$

where ϕ represents the modulator parameters.

3.1.3 Synthesizer

The synthesizer is implemented as a SIREN based model that uses sine activation functions. Its parameters are initialized carefully as proposed in [5], which stabilizes training and preserves high frequency information. The synthesizer receives a spatial coordinate as input and produces the reconstructed function value (here vorticity, ω_t) at that location. This motivated the name synthesizer, as it synthesizes the continuous field from the latent representation.

$$\hat{\omega}_t(x, y) = S_\beta(x, y\,; M_\phi(z_t)) \tag{7}$$

where β denotes the synthesizer parameters.

3.2. Joint Training Objective

The encoder, modulator and synthesizer are trained jointly in tandem, in a self supervised manner. The objective is to reconstruct the input field from its latent embedding. Given the predicted reconstruction $\hat{\omega}_t$, the parameters of all three components are optimized by minimizing the reconstruction loss

$$\min_{\theta,\phi,\beta} \; \|\hat{\omega}_t - \omega_t\|_2^2 \tag{8}$$

The training pipeline therefore learns a mapping from the physical field to a latent space and back to the physical space with minimal reconstruction error.

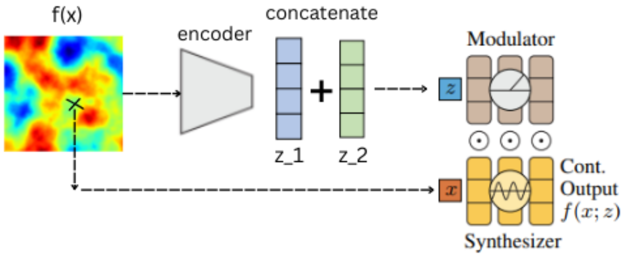


Figure 1. The solution field ω_t ($f(x)$ in the illustration) is encoded into embedding z_1 via a CNN-encoder. z_2 is the implicit embedding, initialized with entries sampled from $\mathcal{N}(0, 0.01)$ and optimized via auto-decoding. The concatenated vector z obtained from z_1, z_2 is fed to the Modulator. The query coordinate (x, y) (x in the illustration) is passed into the synthesizer. The modulator performs scale and shift operations on the intermediate layers of the Synthesizer, which finally predicts the reconstructed field value at the query point.

3.3. Auto-encoder

The auto-encoder setup utilizes all the three components that were presented, and the above joint training objective describes the training formulation of an auto-encoder. Essentially, we have an explicit encoder which outputs the latent embeddings that are used by the two sub-networks to perform the decoding.

3.4. Auto decoder

The earlier auto encoder setup that relied on an explicit CNN encoder has a fundamental limitation. When a CNN is used to compress an input field into a latent space, it naturally focuses on the dominant low frequency structures since most of the energy resides in the lower modes. As a result, the learned latent representation tends to discard finer scale information and the reconstructed outputs inherit this loss. This happens because the CNN encoder learns a single global encoding map that must work for all samples in the dataset, which makes it less suited for representing sample specific high frequency variations.

The auto decoder approach addresses exactly this issue. Instead of using an encoder network, each sample is assigned a randomly initialized and trainable embedding vector. This embedding is optimized so that the reconstruction error is minimized. During inference, the modulator and synthesizer remain fixed and only the embedding vector is optimized. Since the embedding dimension is usually of the order of 10^2 or 10^3, this optimization problem remains lightweight compared to models with 10^5 or 10^6 parameters. This process is known as inference time optimization. Because these embeddings are learned on a per instance basis rather than acting as a global representation, they tend to retain higher modes more faithfully. Training is also significantly faster as the number of trainable parameters is drastically reduced. The tradeoff is that inference can become slower due to the optimization required for each new instance.

3.5. Hybrid approach

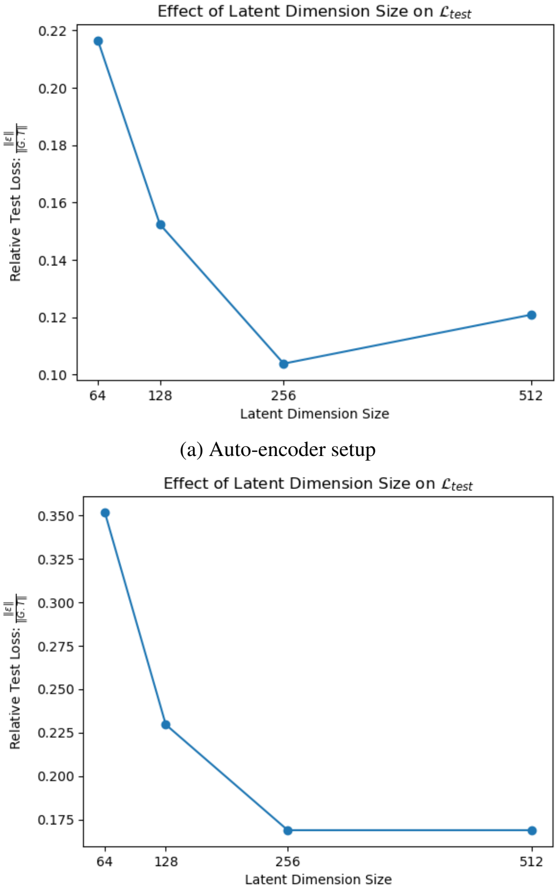
In this setting we combine both components and evaluate their joint performance. We find that merging the CNN encoded latent with the auto decoded embedding improves reconstruction accuracy. The underlying idea is that the CNN encoder captures global structures by exploiting spatial inductive biases, whereas the auto decoded embedding focuses on recovering the unresolved higher modes. These two embeddings complement each other, which motivated us to explore this hybrid formulation. To the best of our knowledge, such a combination has not appeared in the literature. The training process remains largely unchanged except that the modulator now receives a concatenated latent vector. Figure 1 illustrates this pipeline in detail.

4. Results

In this section we present a series of analyses to evaluate the proposed architectures. We begin with an ablation study on the latent dimension to understand its impact on reconstruction accuracy. This is followed by the training loss evolution for both models, illustrating their optimization behaviour. We then present the reconstructed fields using the hybrid approach across multiple time steps to study its ability to capture fine scale dynamics. Finally, we analyse the spectral behaviour of the predictions by comparing their energy spectrum with that of the ground truth.

4.1. Ablation study

We evaluate the effect of latent dimension on both the auto encoder and auto decoder models. As shown in Table 1, increasing the latent size consistently improves the reconstruction quality up to a dimension of 256, after which the performance saturates. We also observe that for the same latent dimension the auto encoder achieves lower reconstruction error compared to the auto decoder, which is expected since the auto encoder learns a global compression map across the dataset whereas the auto decoder relies on per instance optimization. Both methods achieve their best performance at a dimension of 256. Based on this observation, the hybrid model combines the optimal embeddings from both branches and achieves the lowest relative test error, confirming the complementary nature of these two approaches.



(a) Auto-encoder setup

(b) Auto-decoder setup

Figure 2. Ablation study on the latent dimension for both proposed architectures. A latent size of 256 achieves the best tradeoff between reconstruction accuracy and model complexity.

4.2. Loss curves

The training loss curves in Figure 3 provide additional insight into the convergence behaviour of the models. Both the auto encoder and auto decoder are trained for 500 epochs each, using the ReduceLROnPlateau scheduler with an initial learning rate of 5×10^{-3}, reduced by a factor of 0.5 with a patience threshold of 50. The auto encoder shows a systematic decrease in loss, converging to approximately

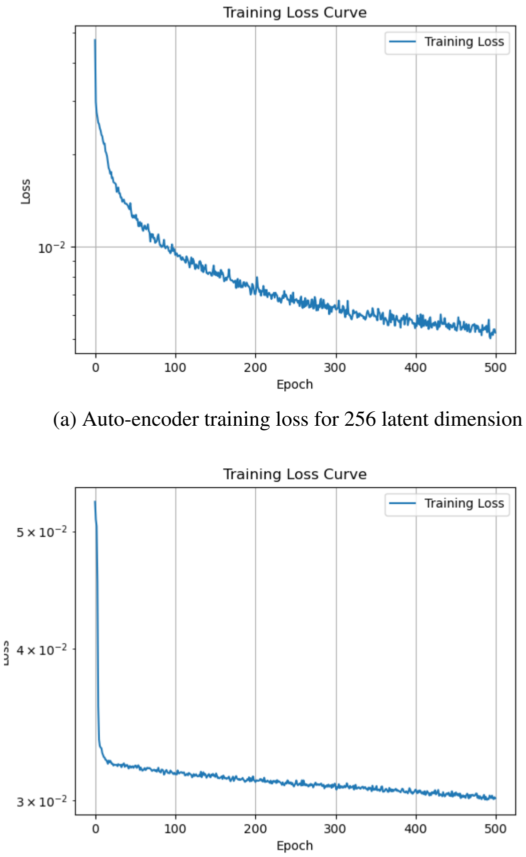
Method	Dimension	Relative test error: $\frac{\|\epsilon\|}{\|G.T\|}$
Auto encoder	64	0.219
	128	0.158
	256	0.103
	512	0.121
Auto decoder	64	0.351
	128	0.228
	256	0.169
	512	0.169
Hybrid	**[256, 256]**	**0.029**

Table 1. Relative test errors for different latent dimensions across all methods. Both auto encoder and auto decoder achieve their best performance at a dimension of 256. The hybrid approach combines these optimal dimensions and achieves the lowest overall error.

5×10^{-3}. In contrast, the auto decoder loss saturates at a higher value around 3×10^{-2}. Overall, both the loss curves show a consistent and systematic reduction in the training loss with increase in epochs. These models can be trained for more number of epochs as the loss curves do not appear to completely saturate at 500 epochs. However due to compute constraints, we limit the training to 500 epochs which by itself yields promising results.

4.3. Reconstructed results

We evaluate the hybrid method by reconstructing the solution fields at multiple time steps $t = 0, 10, 20, 30, 40$ as illustrated in Figure 4. The plotted reconstructions show that the initial states contain stronger high frequency content, while the later states become progressively smoother due to the dissipative nature of the Navier Stokes dynamics. Hence, the reconstruction task is more challenging at early time steps, and the hybrid model does not fully recover the highest modes in these frames. However, as the temporal rollout proceeds and the high frequency energy is damped by the physics, the reconstructions align more closely with the ground truth. This behaviour is consistent across all test samples and suggests that future improvements should focus on enhancing the model's ability to represent sharp features in the early phase of the flow evolution.



(a) Auto-encoder training loss for 256 latent dimension

(b) Auto-decoder training loss for 256 latent dimension

Figure 3. Training loss plots for 500 epochs, comparison across both the methods. We observe significant reduction in training loss for auto-encoder as compared to auto-decoder which stagnates at around 3×10^{-2}.

4.4. Energy spectrum

To study the frequency wise behaviour of the predictions, we compute the isotropic energy spectrum by first applying a two dimensional FFT to both the predicted and ground truth fields. The Fourier coefficients are then grouped into shells defined by $\sqrt{k_x^2 + k_y^2}$, and the energy in each shell is computed. The spectra show close agreement at lower and intermediate modes, indicating that the model captures the dominant structures accurately as presented in Figure 5. At higher wave numbers the predicted spectrum deviates from the ground truth and exhibits a noticeable drop in energy. This reflects the well known spectral bias of neural networks, which tend to under represent high frequency components. Although the hybrid method reduces this discrepancy to some extent, the effect remains visible and highlights an important direction for future work.

5. Challenges

One of the major challenges in this project was setting up a coherent model pipeline. The proposed framework consists of multiple interconnected components that must be trained jointly. Ensuring correct gradient propagation across all sub-networks required careful debugging and validation of the training loop. Even minor inconsistencies in data flow or parameter updates significantly affected model convergence, making this stage the most time-consuming

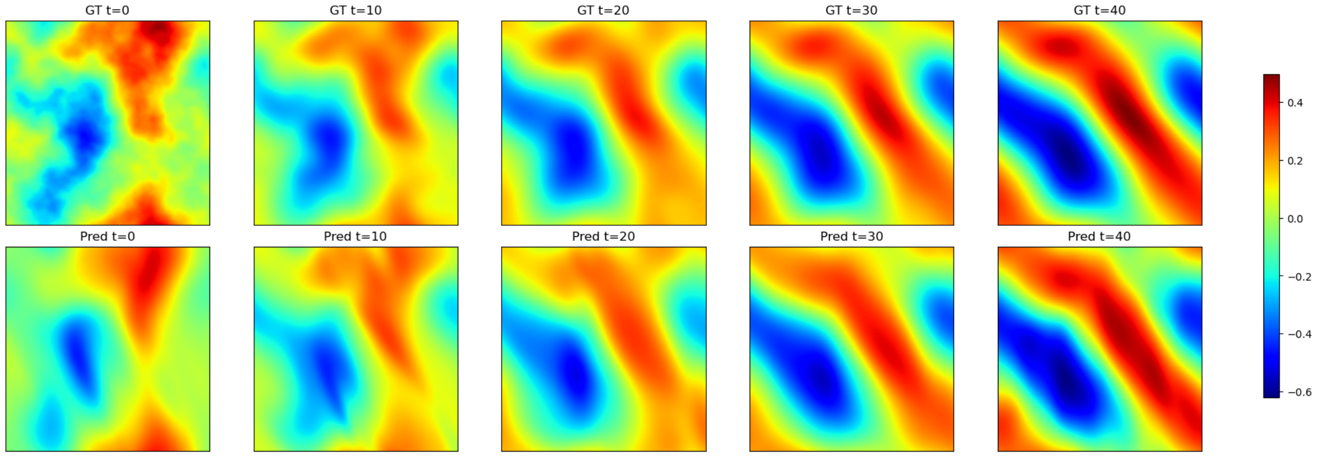
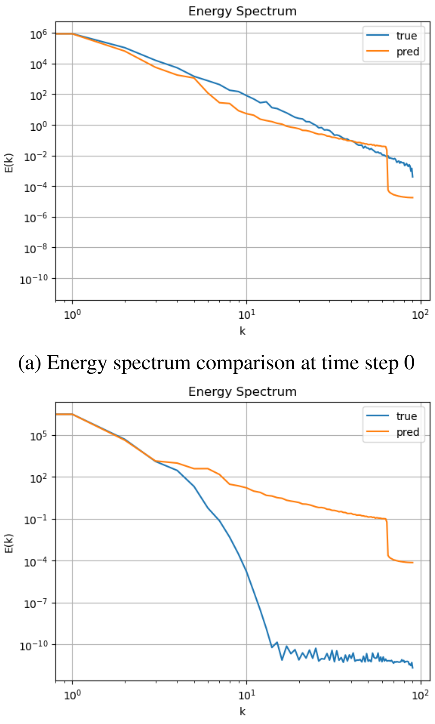


Figure 4. Comparison of ground truth and predicted reconstructed fields across time. Early time steps exhibit richer high-frequency content in the ground truth, which progressively dissipates as the flow evolves. The predicted reconstructions struggle to recover these initial high-frequency modes, a limitation that we aim to investigate further in future work.



(a) Energy spectrum comparison at time step 0

(b) Energy spectrum comparison at time step 40

Figure 5. Comparison of energy-spectrum of the ground truth and the reconstructed field. We observe at t = 0 timestep, the reconstructed field fails to emulate the higher modes. However, for t = 40 time step it overestimates the energy for the higher modes.

aspect of the work.

From the results point of view, a key challenge is the inherent spectral bias which our proposed method also suffers. This bias causes the model to preferentially learn low frequency components while struggling to capture the high frequency modes of the solution field. Spectral bias is an active and unresolved area of research, particularly for Operator Learning. Addressing this with further analysis could lead to meaningful advancements and potentially contribute to the state of the art in Operator Learning.

6. Reflection

This project provided a valuable opportunity to work end-to-end on a topic closely aligned with my research interests. The workflow, from dataset generation and preprocessing to model development, evaluation, and post-processing of results, allowed me to gain a deeper appreciation for the complete research pipeline.

I am grateful to Prof. Chen for granting me the flexibility to pursue a solo project. This freedom enabled me to explore the problem space more broadly, experiment with different ideas, and shape the project according to my curiosity. Overall, the experience has strengthened my interest in neural operator methods and motivated me to further investigate the open challenges encountered in this work.

7. Division of Labor

As this is a solo-project, all the components presented in this project were implemented by myself.

8. Extra Credit

This project was undertaken for extra credit, with all components implemented from scratch, including data generation and post-processing analysis. In my proposal, I had planned to implement a transformer-based sequential model for the time rollout of the latent embeddings. I attempted this approach, but the results were not satisfactory. The primary reason is cascading errors: the latent representations themselves contain inherent inaccuracies, which are further amplified during the autoregressive rollout by the transformer. As a result, the reconstructed physical space outputs are not very accurate. Nonetheless, I am including the transformer implementation with self-attention for reference in my codebase submission. The plan is to pursue time integration once high accuracy in latent reconstruction is achieved.

References

[1] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations, 2021. 1

[2] Bogdan Raonić, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. Convolutional neural operators for robust and accurate learning of pdes, 2023. 1

[3] Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning, 2023. 1

[4] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks, 2019. 1

[5] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions, 2020. 2, 3

[6] Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations, 2021. 2

[7] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation, 2019. 2

[8] 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, 2023. 2

[9] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer, 2017. 3