Deciphering Earth's Physical Dynamics via Multimodal and Low-Dimensional Manifold Learning

Hao Wu[a], Junyuan Mao[d], Jian Zhao[b,c], Qingsong Wen[e], Qing Guo[f], Yang Liu[f], Kun Wang[d,*], Yang Wang[d,*], Xuelong Li[c]

[a]*Machine Learning Platform Department, Tencent, Beijing, 100094, China*
[b]*EVOL Lab, Institute of AI (TeleAI), China Telecom, Beijing, 100032, China*
[c]*School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an, Shaanxi, China*
[d]*(University of Science and Technology of China, Hefei, Anhui, 230022, China*
[e]*Squirrel AI, Seattle, USA*
[f]*Nanyang Technological University, 50 Nanyang Ave, 639798, Singapore*

Abstract

Accurate modeling of physical dynamics is vital for environmental monitoring, climate prediction, and studying human mobility patterns. Advanced hardware technologies like satellite imagery and atmospheric sensors have significantly advanced data-driven deep learning models. However, these models often face issues with efficiency, accuracy, and generalization. We propose a new framework named ManiEarth that combines Large Language Models (LLMs) with manifold learning to overcome these challenges. LLMs enhance generalization by effectively integrating multimodal data, while manifold learning improves efficiency by uncovering low-dimensional intrinsic features. The framework uses Fourier Neural Operator (FNO) layers and leverages text prompts for physical equations, external forces, and boundary conditions to develop robust feature representations. Our approach addresses limitations in data utilization and computational efficiency, resulting in more practical and robust models. Experimental evaluations show significant improvements in training and inference speed, prediction accuracy, and adaptability in varied environmental conditions. For example, in modeling complex dynamic systems, our model achieves an error of 0.2465 on the Navier-Stokes equations, demonstrating clear advantages over existing models.

Keywords: Physical dynamics, Large Language Models, Manifold learning, Multimodal data integration

1. Introduction

Modeling the evolution of physical dynamics lays a foundational framework for analyzing and forecasting physical systems [1, 2, 3, 4], encompassing significant natural and social science research topics such as environmental modeling [5, 6], climate science [7, 8], and human mobility studies [9, 10]. In recent years, the widespread adoption of hardware technologies such as satellite imagery and atmospheric sensors has facilitated the storage and utilization of high-quality multimodal information [11], including text, images, and video streams. This technological advancement has significantly propelled the rapid development of data-driven deep learning models, enabling more sophisticated and accurate predictive capabilities across diverse fields.

Generally, modeling physical dynamical systems can be regarded as a spatio-temporal forecasting problem [12, 13],

*Corresponding author. Hao Wu, Junyuan Mao and Jian Zhao are contributed equally to this work.
Email addresses: wuhao2022@mail.ustc.edu.cn (Hao Wu), maojunyuan@mail.ustc.edu.cn (Junyuan Mao), wk520529@mail.ustc.edu.cn (Kun Wang), angyan@ustc.edu.cn (Yang Wang), xuelong_li@chinatelecom.cn (Xuelong Li)

involving changes in system states over time and space. Key factors include temporal evolution, spatial distribution, and the coupling of both. Upon reviewing deep learning methodologies in this area, the principal approaches for tackling the spatio-temporal modeling of physical dynamical systems can be categorized into three main types: visual backbone models, operator learning models, and Physics-Informed Neural Networks (PINNs). Visual backbone models, such as convolutional neural networks [14, 15] and Transformer models [16, 3], extract features from spatio-temporal data and are suitable for handling continuous snapshot data of dynamical systems. Operator learning models, like Neural Operators, learn mapping relationships in continuous function spaces [6, 17, 18, 19, 20], making them suitable for solving partial differential equations (PDEs). PINNs directly incorporate physical constraints into neural networks [21], enhancing physical consistency and predictive accuracy by minimizing physical residuals. However, PINNs, which are a type of neural network that incorporates physical laws expressed as partial differential equations into the training process (usually adding a loss constraint [22]), often introduce issues with weak generalization and inefficient training. As such, studies focusing on these aspects fall outside the scope of this paper.

Despite achievements in modeling spatio-temporal physical dynamical systems [23, 3, 20, 6], off-the-shelf technologies have clear limitations [24, 25]. Most deep learning models lack efficiency and accuracy in utilizing observational data. Common issues, such as over-smoothing, prevent models from fully exploiting potential information, thereby affecting prediction accuracy [26, 27]. Additionally, these models often focus on a single data source when handling multimodal information, ignoring interactions between different modalities and missing rich physical parameters like boundary conditions, viscosity coefficients, and forcing terms [6, 28].

Existing tailor-made physical dynamics modeling architectures, such as RNNs [14, 29], CNNs [2, 30], and Transformers [3, 26], face significant computational burdens and efficiency bottlenecks when processing high-resolution spatio-temporal data. These limitations hinder their training and inference speed, reducing their practicality and applicability in real-time scenarios. Furthermore, these models typically fail to meet practical needs for high-quality data and lack sufficient generalization capabilities, making it difficult to adapt to diverse and changing environments and conditions. In summary, current technologies in spatio-temporal physical dynamical systems suffer from inefficiency, high computational costs, and insufficient generalization abilities, urgently requiring new methods and technologies to address these challenges.

In fact, text is an important perceptual channel that provides rich context and semantic information, crucial for integrating and understanding multimodal data [31, 32]. Current large language models (LLMs), such as GPT-4 [33] and BERT [34], excel in text understanding and have become the main tools for handling natural language tasks. They not only comprehend and generate complex language structures but also extract and integrate information from multimodal data [35, 36], enhancing the models' generalization ability. By capturing the complex relationships between text and other modalities, such as images, audio, and video, LLMs generate more robust feature representations. This multimodal fusion capability allows the models to perform better in practical applications with diverse data, improving prediction accuracy and adaptability in various environments [37, 38, 39]. For example, in environmental modeling and climate science [40], LLMs can combine textual data, like weather reports and research papers, with image data, such as satellite images and meteorological charts, to provide more comprehensive predictive and analytical capabilities [41]. In human mobility studies [42], LLMs can combine social media text data with location data to offer deeper insights into behavioral patterns. Overall, the outstanding performance of LLMs in text understanding and multimodal data integration makes them powerful tools for solving complex prediction and analysis tasks.

Going beyond this perspective, currently popular manifold learning shows significant advantages in handling high-dimensional complex data [20, 43]. It embeds high-dimensional data into low-dimensional manifolds, preserving the core structure and key information of the data, thus effectively reducing the computational burden [44]. This method is particularly suitable for processing physical system data [45], which often features high dimensionality and complexity. Through manifold learning, we can uncover the low-dimensional essential features in the data, enhancing the efficiency of model training and inference, and significantly improving real-time performance and applicability [46]. Additionally, manifold learning

maintains the topological structure of the data during embedding, making the model more robust and accurate in predictions [47, 20]. It not only reduces computational costs but also enhances model performance in processing high-resolution spatio-temporal data, offering stronger adaptability to handle complex changes in various practical application scenarios. For example, in climate science and environmental modeling, manifold learning helps extract key physical features, thus boosting the model's predictive capability and decision support level. Overall, manifold learning provides a powerful tool for complex data modeling through efficient feature extraction and dimensionality reduction, solving current technological bottlenecks in data utilization and computational efficiency, and enhancing the model's feasibility and robustness.

Inspired by the aforementioned techniques, we have proposed a novel method that combines manifold learning and large language models. By combining the strengths of LLMs [48] and manifold learning [20, 44, 43], we have built an efficient and accurate spatio-temporal physical dynamics system model. The model first uses convolutional neural networks to encode input data and extract key features. Then, it employs spectral neural operator layers (FNO layers) [6] to learn and retain the data's low-frequency information, and uses manifold learning to map high-dimensional data to a low-dimensional space, preserving its core structure and important information. This step significantly reduces the computational load and enhances the model's efficiency and real-time performance. Moreover, the model integrates the text processing capabilities of LLMs, extracting physical equations and conditions from textual data through understanding and generating complex language structures [49], and incorporating this information into the model. Through deep-wise Conv2d, these physical insights are converted into a format directly usable by the model, further enhancing its physical consistency and predictive accuracy. Our approach significantly improves the model's accuracy and efficiency, providing more reliable predictions in diverse environments and conditions. By combining the generalization ability of LLMs with the feature extraction of manifold learning, our model achieves more efficient and accurate predictions in a wide range of applications, addressing existing technological bottlenecks in data utilization and computational efficiency, and enhancing the model's feasibility and robustness.

Specifically, our contributions are as follows. ❶ **Multimodal Data Integration**:We use LLMs and manifold learning to effectively integrate text with other modal data, improving the model's feature representation capability. ❷ **Enhanced Physical Consistency**: By introducing physical constraints and conditions, we enhance the model's consistency and accuracy in predicting physical dynamic systems. ❸ **Efficient Computation**: We leverage manifold learning and spectral neural operator layers to reduce the computational complexity of high-dimensional data, increasing the model's training and inference speed, making it suitable for processing high-resolution spatiotemporal data in practical applications.

2. Related work

2.1. Deep Learning for Physical Dynamics

In recent years, many resilient deep learning methods have been developed for learning physical dynamics. Due to similar tensor shapes, modeling physical systems is viewed as a computer vision problem. Several state-of-the-art models designed for computer vision tasks (e.g., image super-resolution or video prediction) are used to model physical dynamics [23, 3]. However, physics inconsistency, unexplainability, and poor generalization limit further development. More deep-learning methods guided by physics theory have been designed, which can be roughly categorized into equation-constraint, interpretable-structure, and operator learning methods.

Equation-constraint methods. Incorporating physical laws into the loss function, physics-informed machine learning methods [13] ensure that the prediction results satisfy specific physical properties, even achieving unsupervised prediction for equation-governed dynamics [50, 51, 52, 53, 54, 55]. However, in complex scenarios such as real-world dynamics, incomplete physical laws and the imbalance of loss function terms make optimization for neural networks significantly difficult, thereby limiting the performance of the models [56, 57].

Interpretable-structure methods. Interpretable-structure methods for learning physical dynamics use the mathematical equivariance between deep learning structures and physical equations to design architectures that incorporate more physical inductive bias. PDE-Net [58] demonstrates the similar mathematical properties of the convolution operation as the difference operator and leverages this theory to develop a framework for learning

time-dependent partial differential equations. Neural ODE [59] shows that continuous Residual Networks [60] can be mathematically expressed as ordinary differential equations and are utilized for predicting dynamic systems [17, 61, 62]. Based on Noether's theorem, equivariant deep learning methods incorporate geometric symmetry into neural networks through equivariant group transformations for constraining the conservation of physical systems [63, 64, 65, 66, 67]. However, such models with strong physics inductive bias in their structure may not be generically applicable in different physics scenarios and may even degrade the performance and generalization capability of the model in real-world dynamics with noisy or incomplete data [68].

Operator-learning methods. Operator learning methods are designed to learn mappings between infinite-dimensional function spaces. Based on the universal approximation theorem [69, 70], DeepONet [71] learns the target operator by sampling the function space. The Koopman theory [72] inspires several methods aimed at approximating the infinite-dimensional Koopman operator in the observation space [25, 73, 74, 75]. Additionally, Green's function-based models convert infinite-dimensional operator mappings into kernel integral parameterization [6, 18, 76, 77, 78, 79]. However, these methods learn the underlying operators in a high-dimensional latent space, but further discussion about the dimension of the latent space is still lacking. Accurately representing infinite-dimensional operators in a finite-dimensional parameter space remains a challenge.

2.2. Dimension Representation of Operators

The intrinsic dimension can be conceptualized as the minimum number of variables or parameters required for the minimal representation, often regarded as the minimal number of hidden neurons necessary for a deep learning model to represent the target. Estimating the intrinsic dimension of physical systems is beneficial for understanding the underlying intrinsic dynamics behind the data [80, 81], finding parameterized surrogate models, and building reduced order models [82, 83, 84]. For operator-learning methods, finite-dimensional representations for infinite operators are indispensable. The Low-rank Neural Operator [18] reconstructs r-rank operators using SVD. Dynamic Mode Decomposition [85, 86] and several Koopman-based deep learning methods [73, 74, 75] have been developed to identify the invariant subspace of the Koopman operator, allowing finite-dimensional linear representations of complex dynamic systems. NOMAD learns a low-dimensional representation of the solution using a nonlinear manifold decoder [87]. With a new universal approximation theorem under minimal assumptions for the underlying operator, PCA-Net partially overcomes the general curse of dimensionality for operator learning [88]. However, there still lacks a unified paradigm designed for learning the intrinsic dimensional representation of operators applicable to various physical systems and model structures.

3. Method

3.1. Overview

Our method combines Large Language Models (LLMs) with manifold learning to address challenges in modeling physical dynamics. This innovative framework integrates multimodal data through LLMs and utilizes manifold learning to uncover low-dimensional intrinsic features, enhancing both efficiency and accuracy. The model employs Fourier Neural Operator (FNO) layers to extract features in frequency domains and leverages text prompts in LLMs for physical equations, external forces, and boundary conditions to develop robust feature representations. By merging these technologies, we achieve significant improvements in training and inference speed, prediction accuracy, and adaptability in diverse environmental conditions.

The integration of LLMs allows the model to understand and process complex language structures, extracting valuable information from textual data. This information is then used to refine the model's predictions, ensuring that the physical dynamics are accurately captured. The use of manifold learning further enhances the model by reducing the dimensionality of the data, making the computations more efficient without losing essential features.

Problem Definition To provide a clear understanding, we elucidate the key concepts involved. Consider a video sequence that captures a dynamic physical system over time, consisting of T time steps, denoted as $V_{1:T} = \{V_1, \ldots, V_T\}$. Each frame records C color measurements across all points within a spatial region represented by an $H \times W$ grid. From a spatial perspective, the observation of these C measurements at any given time step i

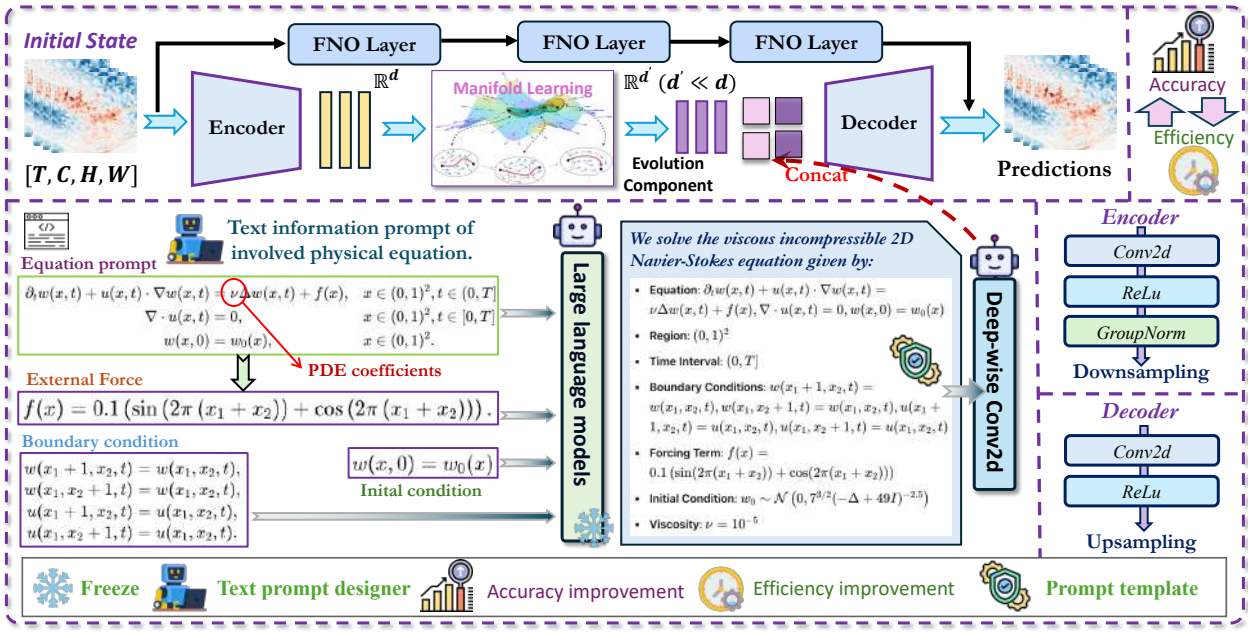

Figure 1: **Overview of the framework.** The figure illustrates the proposed framework for modeling physical dynamics using a combination of Large Language Models (LLMs) and manifold learning. The initial state, represented as spatio-temporal data $[T, C, H, W]$, is processed through an encoder that extracts high-dimensional features. These features are then passed through multiple Fourier Neural Operator (FNO) layers and a manifold learning component to capture temporal and spatial dependencies, reducing the dimensionality to $\mathbb{R}^{d'}$. Simultaneously, text information prompts are used to encode physical equations, external forces, and boundary conditions, which are processed by LLMs to enhance the model's understanding and integration of physical constraints. These textual features are further refined using deep-wise Conv2d layers. The outputs from the manifold learning and LLM components are concatenated and fed into a decoder, which reconstructs the predictions.

can be represented as a tensor, $V_i \in \mathbb{R}^{C\times H\times W}$. Our goal is to use spatio-temporal data to uncover underlying physical principles by effectively integrating multimodal data and uncovering low-dimensional intrinsic features to forecast the most probable future sequence of length T_f, denoted as $V_{T+1:T+T_f} = \{V_{T+1}, \ldots, V_{T+T_f}\}$.

3.2. Learning information from Equations via LLMs

LLMs are integrated into our framework to enhance generalization by effectively handling multimodal data. Text prompts are used to encode physical equations, external forces, and boundary conditions. These prompts are processed by LLMs to generate a structured representation of the problem, which is then used to guide the model. And then some features of texts are extracted by Conv2d. LLMs extract and integrate this information into the model, enhancing the physical consistency and predictive accuracy. The text prompt designe ensures that the prompts are accurately formulated and integrated into the model. Then, we share the detailed processes.

3.2.1. Equation Prompt

The equation prompt encodes the governing physical equations, such as the viscous incompressible 2D Navier-Stokes equation. The encoded process is represented as:

$$\partial_t w(x,t) + u(x,t)\cdot\nabla w(x,t) = \nu\Delta w(x,t) + f(x) \quad (1)$$

where $w(x,t)$ is the vorticity, $u(x,t)$ is the velocity field, ν is the viscosity, Δ is the Laplacian operator, and $f(x)$ is the forcing term. These prompts are designed using a Text Prompt Designer to ensure precise and accurate representation of the physical equations.

By encoding these equations, the model can better understand the underlying physical processes, leading to more accurate and reliable predictions. This step is crucial for

ensuring that the model's predictions are physically plausible and consistent with known scientific principles. The text prompt designer plays a vital role in ensuring that the encoded equations are accurate and relevant, thus enhancing the model's interpretability and reliability.

3.2.2. Boundary Condition

The boundary conditions are encoded to maintain the physical constraints at the boundaries of the spatial domain. This is crucial for ensuring the predictions are physically plausible. An example boundary condition is represented as:

$$w(x_1 + 1, x_2, t) = w(x_1, x_2, t), \text{ for } x_1, x_2 \in [0, 1], t \in [0, T] \tag{2}$$

Encoding boundary conditions helps the model to respect the physical constraints imposed by the environment, ensuring that the predictions do not violate any fundamental laws of physics. This adds an extra layer of robustness to the model. Properly encoded boundary conditions ensure that the model's predictions are consistent with the physical realities of the system being modeled.

3.2.3. External Force

External forces acting on the system are encoded to account for influences that drive the dynamics. An example external force is represented as:

$$f(x) = 0.1\,(\sin(2\pi(x_1 + x_2)) + \cos(2\pi(x_1 + x_2))) \tag{3}$$

By including external forces in the model, we can account for additional factors that influence the system's behavior. This ensures that the model's predictions are comprehensive and take into account all relevant influences. External forces are often critical in driving the dynamics of physical systems, and their accurate representation is essential for reliable predictions.

3.2.4. Deep-wise Conv2d with Prompt Template

After processing by LLMs, the inputs are further refined using deep-wise convolutional neural networks (Conv2d) to extract deep features that are essential for accurate predictions. This process utilizes standardized Prompt Templates to ensure consistency and efficiency. In this module, we leverage LLMs, specifically GPT, to extract features from physical equations, external forces, and boundary conditions. These extracted textual features are then encoded and processed using deep-wise convolutional neural networks (Conv2d) to further refine the information.The encoded text information, represented as $\mathbf{T} \in \mathbb{R}^{L \times d}$, where L is the length of the text and d is the feature dimension, is then fed into the deep-wise Conv2d layers.The deep-wise Conv2d layers are responsible for extracting and refining features from the encoded textual information. Depth-wise convolution is applied to each channel separately, followed by pointwise convolution to combine features across different channels.And an upsampling operation is included within the deep-wise Conv2d layers to increase the spatial resolution of the feature maps, ensuring that the dimensions are suitable for final integration and prediction. This process helps the model generate high-resolution predictions that are both detailed and accurate and can be expressed as:

$$\mathbf{F} = \text{ReLU}(\text{Conv2d}(\text{Upsample}(\mathbf{T}))) \tag{4}$$

where $\mathbf{T}$ is the encoded textual input, Upsample denotes the upsampling operation, Conv2d represents the convolutional operation, ReLU is the activation function, and $\mathbf{F} \in \mathbb{R}^{d'}$ is the resulting low-dimensional feature representation.

3.3. Extracting and Preserving Essential Features from videos

We start by encoding the initial state of the sequence as a discrete tensor, transforming the data into a suitable format for processing. The encoder utilizes convolutional neural networks to extract spatial features, which are then reduced in dimensionality through manifold learning. Fourier Neural Operator (FNO) layers capture the evolution of these features in the frequency domain, allowing for efficient learning of underlying dynamics. And the decoder reconstructs the output by restoring spatial dimensions. The final predictions are generated by combining the outputs from the FNO layers with those of the decoder. Then, we introduce our FPG module in detail.

3.3.1. Encoder

The encoder utilizes convolutional neural networks (Conv2d) to extract spatial features from the input data. Each time step i is expressed as $V_i \in \mathbb{R}^{C \times H \times W}$, and

the encoded tensor is transformed into a relatively high-dimensional representation $X_d \in \mathbb{R}^d$. The encoding process is represented as:

$$E(x) = \text{ReLU}(\text{GroupNorm}(\text{Conv2d}(x))) \tag{5}$$

where x is the input tensor, Conv2d denotes the convolutional operation, GroupNorm is the group normalization, and ReLU is the rectified linear unit activation function.

The encoder's role is pivotal as it translates raw input data into a form that can be effectively utilized by subsequent layers. By extracting spatial features through convolutional layers, the encoder captures local patterns and structures within the data, which are essential for understanding the physical system's behavior. This high-dimensional representation X_d serves as a comprehensive descriptor of the input state, preserving critical information while facilitating efficient processing.

3.3.2. Manifold Learning

Manifold learning further reduces the dimensionality of the data while preserving its essential features. This process maps the encoded data X_d to a low-dimensional manifold $Y_{d'}$, where $d' \ll d$:

$$Y = \phi(X) \tag{6}$$

where ϕ is the nonlinear mapping function learned during training. This mapping helps in maintaining the intrinsic structure of the data, which is crucial for accurate predictions.

By reducing the dimensionality, manifold learning makes the computations more efficient, allowing the model to process large amounts of data in a reasonable time frame. This reduction is particularly important for high-dimensional data, where computational complexity can be a significant challenge. Manifold learning techniques enable the model to focus on the most relevant aspects of the data, effectively filtering out noise and redundant information, thus enhancing both accuracy and efficiency.

3.3.3. Fourier Neural Operator (FNO) Layers

The Fourier Neural Operator layers capture the evolution of the data in the frequency domain. The transformation within an FNO layer is given by:

$$\mathcal{F}(u)(k) = \hat{u}(k) = \int_\Omega u(x) e^{-2\pi i k \cdot x} \, dx \tag{7}$$

where $\mathcal{F}$ denotes the Fourier transform, $u(x)$ is the input function, k is the frequency component, and $\hat{u}(k)$ represents the transformed function. The FNO layer processes $\hat{u}(k)$ through learned filters and applies an inverse Fourier transform to produce the output:

$$u'(x) = \mathcal{F}^{-1}(\hat{u}(k) \cdot W(k)) \tag{8}$$

Here, $W(k)$ represents the learned weights in the frequency domain, and $\mathcal{F}^{-1}$ is the inverse Fourier transform.

The use of FNO layers allows the model to efficiently capture the evolution of the data in the frequency domain, leading to more accurate predictions of the physical dynamics. This approach leverages the power of Fourier transforms to handle complex temporal dependencies in the data. By working in the frequency domain, the model can identify and utilize patterns that might be less apparent in the time domain, enhancing its ability to predict future states accurately.

3.4. Integration Features from videos and texts

The refined inputs from the LLM module and the outputs from the manifold learning module are integrated to form the final prediction. Mathematically, this can be expressed as follows:

$$P = \sigma\left(W_1 \cdot \text{Decoder}(Y_{d'}) + W_2 \cdot \text{LLM}(\text{Prompt})\right) \tag{9}$$

where P represents the final prediction, σ is an activation function, W_1 and W_2 are learnable weights, $\text{Decoder}(Y_{d'})$ is the output after manifold learning, and LLM(Prompt) is the output processed by the LLMs.

By incorporating the Large Language Models Module, our framework effectively combines data-driven approaches with physical laws, enhancing both the efficiency and accuracy of the predictive model. This integrated approach supports robust solutions for complex physical systems, demonstrating significant potential for solving partial differential equations (PDEs). The final integration step ensures that the predictions leverage both the spatial features and the learned physical constraints, resulting in a comprehensive and accurate model output.

3.5. Predictions

The decoder reconstructs the output from the low-dimensional manifold representation using convolutional

layers and upsampling techniques. The decoding process is expressed as:

$$D(y) = \text{ReLU}(\text{Conv2d}(\text{Upsample}(y))) \tag{10}$$

where y is the low-dimensional input, Upsample denotes the upsampling operation, and $D(y)$ is the reconstructed output.

This reconstruction process ensures that the final output retains the essential features of the original data, while also being in a format suitable for interpretation and analysis. The use of upsampling techniques helps in restoring the spatial resolution of the data, making the predictions more accurate and detailed. The decoder's ability to reconstruct high-quality outputs from low-dimensional representations highlights the effectiveness of the manifold learning module in preserving crucial information.

4. Experiments

We evaluate our method in extensive experiments. Specifically, we conduct experiments on four benchmarks and compare it with six baselines. We aim to address the following **R**esearch **Q**uestions (**RQ**):

RQ1. Does our method consistently outperform all comparison models?

RQ2. How efficient is our method in terms of predictive performance?

RQ3. Does our method effectively capture physical consistency?

RQ4. Can our method generalize to different parameter setting?

4.1. Benchmarks & Baselines

Navier-Stokes Equation. The Navier-Stokes equation models the dynamics and mass transfer of general fluids. We use a two-dimensional equation for an incompressible viscous fluid with a viscosity coefficient of 10^{-5} to test the model's ability to learn complex fluid dynamics at high Reynolds numbers. The dataset is computed using the pseudospectral method, with the vorticity form of the equation as follows:

$$\begin{cases} \partial_t w(x,t) + u(x,t) \cdot \nabla w(x,t) = \nu \Delta w(x,t) + f(x), \\ \nabla \cdot u(x,t) = 0, \\ w(x,0) = w_0(x), \end{cases} \tag{11}$$

The forcing term is fixed at $f(x) = 0.1(\sin(2\pi(x_1 + x_2)) + \cos(2\pi(x_1 + x_2)))$, with initial conditions generated by $w_0 \sim \mu$, where $\mu = \mathcal{N}(0, 7^{3/2}(-\Delta + 49I)^{-2.5})$, and periodic boundary conditions.

Shallow Water Equations. The shallow water equations describe fluid flow in a shallow water approximation and barotropic systems, commonly used for large-scale geophysical flows and tsunami simulations. This dataset is ideal for testing mass conservation and long-term forecasting performance. The equations are expressed as:

$$\begin{cases} \partial_t h + \partial_x(hu_x) + \partial_y(hu_y) = 0 \\ \partial_t(hu_x) + \partial_x\left(u_x^2 h + \frac{1}{2} g_r h^2\right) = -g_r h \partial_x b \\ \partial_t(hu_y) + \partial_y\left(u_y^2 h + \frac{1}{2} g_r h^2\right) = -g_r h \partial_y b \end{cases} \tag{12}$$

Here, h represents the water column depth, u_x and u_y are the velocity components along the x and y axes, respectively, b denotes the seabed topography changes, and g_r is the gravitational acceleration.

Rayleigh-Bénard Convection. The Rayleigh-Bénard convection equations model turbulence caused by convection due to bottom heating, a primary mechanism in El Niño and Southern Oscillation. This dataset is simulated using the lattice Boltzmann method, suitable for testing models' ability to learn turbulence and energy conservation. The equations are:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} + (1 - \alpha(T - T_0))\mathbf{X} \\ \frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla)T = \kappa \Delta T \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \tag{13}$$

Here, $\mathbf{u}$ is the velocity field, T is temperature, α is the thermal expansion coefficient, and κ is the thermal conductivity coefficient.

Reaction-Diffusion Equations. The reaction-diffusion equations simulate the interaction between material diffusion and chemical reactions, commonly describing processes in materials, biology, and the environment. This dataset is computed using a standard finite volume solver and challenges benchmarking due to two nonlinearly coupled variables (activator and inhibitor). The equations are:

$$\begin{cases} \partial_t u = D_u \partial_{xx} u + D_u \partial_{yy} u + R_u \\ \partial_t v = D_v \partial_{xx} v + D_v \partial_{yy} v + R_v \end{cases} \tag{14}$$

The diffusion coefficients are $D_u = 1 \times 10^{-3}$ and $D_v =$

5×10^{-3}, with the reaction functions defined as:

$$\begin{cases} R_u(u,v) = u - u^3 - k - v \\ R_v(u,v) = u - v \end{cases} \tag{15}$$

The constant $k = 5 \times 10^{-3}$.

Baselines. We evaluate ManiEarth against 6 notable models across three benchmarks, dividing them into four categories: ▹ **Visual Backbone Networks.** like U-Net [89], ResNet [60] and Swin Transformer [90]; ▹ **Operator learning methods** with FNO [6], F-FNO [91], and LSM [5].

4.2. Metric

Statistical metrics enable us to evaluate model performance at the pixel level. However, to determine if the model accurately captures physical properties rather than just achieving local fitting, it is crucial to incorporate physical metrics. Thus, we employ the following metrics for evaluation: in addition to the Root Mean Square Error (RMSE) as a statistical metric, the remaining metrics are physical.

- **Root Mean Square Error(RMSE)** is a widely accepted metric for quantifying the statistical performance of the deep learning model, which can reflect the average error of the prediction.

- **Mass Conservation.** For incompressible shallow water wave equation with free surface and closed boundary, the prediction variable h not only describes the depth of water but also is proportional to the mass of the water column. Therefore, the total mass of the system can be calculated by the variable h to evaluate whether the models preserve first-order conserved quantities. The mass conservation formula of the 2-dimensional shallow water equations can be expressed as

$$\frac{d}{dt}\iint_D h dx dy = 0, \tag{16}$$

  where D is the computational domain of the equations in 2-dimensional Euclidean space $\mathbb{R}^2$.

- **Energy Conservation.** For the Rayleigh-Bénard convection scenario, the turbulence energy spectrum indicates the kinetic energy contained in eddies with wavenumber k. The turbulence energy spectrum is calculated by mean turbulence kinetic energy after Fourier transformation. The metric is appropriate for analyzing energy consistency in different ranges of wavenumber. The energy spectral $E(k)$ is calculated by

$$\int_0^\infty E(k)dk = \frac{1}{2T}\sum_{t=0}^{T}[(u_x(t)-\bar{u_x})^2+(u_y(t)-\bar{u_y})^2], \tag{17}$$

  where u_x and u_y are the components of velocity for the x-axis and y-axis, the bar symbol means time average, t is the time step and T denotes the prediction length.

- **Divergence.** Derived by the continuity equation, the divergence of velocity $\nabla \cdot \mathbf{u}$ should be zero for the incompressible fluid parcel, which is the closure condition and fundamental constraint of mass conservation in fluid dynamics. Calculating the average of absolute divergence in the whole fluid field as a physical metric at each time step indicates whether the model learns the intrinsic dynamics of fluid transportation. The divergence formula can be expressed as

$$\nabla \cdot \mathbf{u} = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} = 0, \tag{18}$$

  where $\mathbf{u} = \{u_x, u_y\}$ is a 2-dimensional velocity vector.

4.3. Overall Performance (RQ1)

We summarize the results in Table 1 and Figure 2, from the experimental results, We make the following **Obs**ervations:

Obs 1. Table 1 shows the performance comparison of six baseline models in different scenarios, focusing on the performance advantages of our method over others. When comparing the six baseline models under known and unknown equations, our method shows significant advantages. Specifically, in the context of unknown equations, our method achieves an error of 0.1608 on the SEVIR dataset, which is much better than U-Net's 2.0280 and ResNet's 2.0787, and also far surpasses Swin-Transformer and FNO. This demonstrates that our model has stronger generalization ability and prediction accuracy in handling complex and unknown dynamical systems. On the Kuroshio dataset, our method also performs excellently, achieving an error of 0.0399, outperforming

Table 1: Performance comparison of six baseline models across different scenarios. The scenarios include unknown equations (SEVIR, Kuroshio, Typhoon) and known equations (Navier-Stokes, Shallow Water, Rayleigh-Bénard Convection, Diffusion Reaction). Our model shows superior prediction accuracy in all tested scenarios.

Model	Unknown Equations			Known Equations			
	SEVIR	Kuroshio	Typhoon	Navier Stokes	Shallow Water	Rayleigh-Bénard Convection	Diffusion Reaction
U-Net	2.0280	0.0591	0.0546	0.4451	0.0890	0.3977	0.0612
ResNet	2.0787	0.0709	0.1246	0.5246	0.0730	0.5746	0.0820
Swin-Transformer	2.0067	0.1682	0.0273	0.4741	0.0434	1.6852	*
FNO	1.0099	0.4432	0.1455	0.2547	0.0045	0.5433	0.0008
FNO	1.3455	0.4763	0.1398	0.3211	0.0092	0.5213	0.0012
LSM	1.2569	0.4509	0.1391	0.2863	0.0087	0.5093	0.0009
ManiEarth	**0.1608**	**0.0399**	**0.0152**	**0.2465**	**0.0025**	**0.1392**	**0.0007**

Table 2: Comparative analysis of various models in terms of computational efficiency and performance metrics.

	FLOPs (M)	Param (M)	Inference (s)	MSE	SSIM
ManiEarth	**571.42**	**0.84**	**0.64**	**30.05**	**0.99**
ResNet	686.36	9.68	0.68	30.32	0.98
FNO	743.15	38.71	0.87	30.11	0.98
ManiEarth	**288.79**	**18.24**	**0.71**	**21.15**	**0.97**
ResNet	652.40	50.10	0.85	25.16	0.95
FNO	591.25	40.38	0.83	22.64	0.96

U-Net (0.0591), ResNet (0.0709), and other baseline models. This highlights our method's exceptional capability in complex domains such as ocean current prediction. On the Typhoon dataset, our method significantly leads with an error of 0.0152, outperforming all other baseline models. This not only shows the advantage of our method in handling meteorological data but also further validates its reliability and accuracy in predicting extreme weather events. For known equations, our method performs excellently in classical physical scenarios such as Navier Stokes, Shallow Water, Rayleigh-Bénard convection, and Diffusion Reaction. In predicting the Navier Stokes equations, our method achieves an error of 0.2465, slightly better than FNO's 0.2547. In predicting Shallow Water, our method leads all other models with an error of 0.0025, showing its outstanding performance in fluid dynamics simulations. In Rayleigh-Bénard convection and Diffusion Reaction scenarios, our method achieves errors of 0.1392 and 0.0007, significantly better than all other baseline models. These results fully demonstrate our method's excellent adaptability and accuracy in handling complex physical equations and dynamic systems. In summary, our method performs excellently in various scenarios and datasets, showing clear advantages and superiority in predicting dynamical systems with both unknown and known equations. The comprehensive comparison with multiple baseline models fully proves the wide applicability and superior performance of our method in various fields.

In Figure 2, our method demonstrates outstanding performance across different physical equations, showing significant advantages over other baseline models. In the predictions for Rayleigh-Bénard convection, Navier-Stokes equations, shallow water equations, and diffusion-reaction equations, our method accurately captures the trends in the true values. Particularly at the final time step, our predictions closely match the true values, indicating that our model has superior generalization ability and prediction accuracy for complex physical dynamic systems. These results validate the effectiveness of our approach in multimodal data integration and low-dimensional manifold learning

4.4. ManiEarth Consistently Leading in Efficiency (RQ2)

When comparing the performance of different models in Table 2, ManiEarth shows a significant efficiency advantage. Here are the specifics: ManiEarth's floating point operations (FLOPs) are 571.42M and 288.79M, parameter count (Param) are 0.84M and 18.24M, and inference times are 0.64s and 0.71s, respectively. In contrast, ResNet's

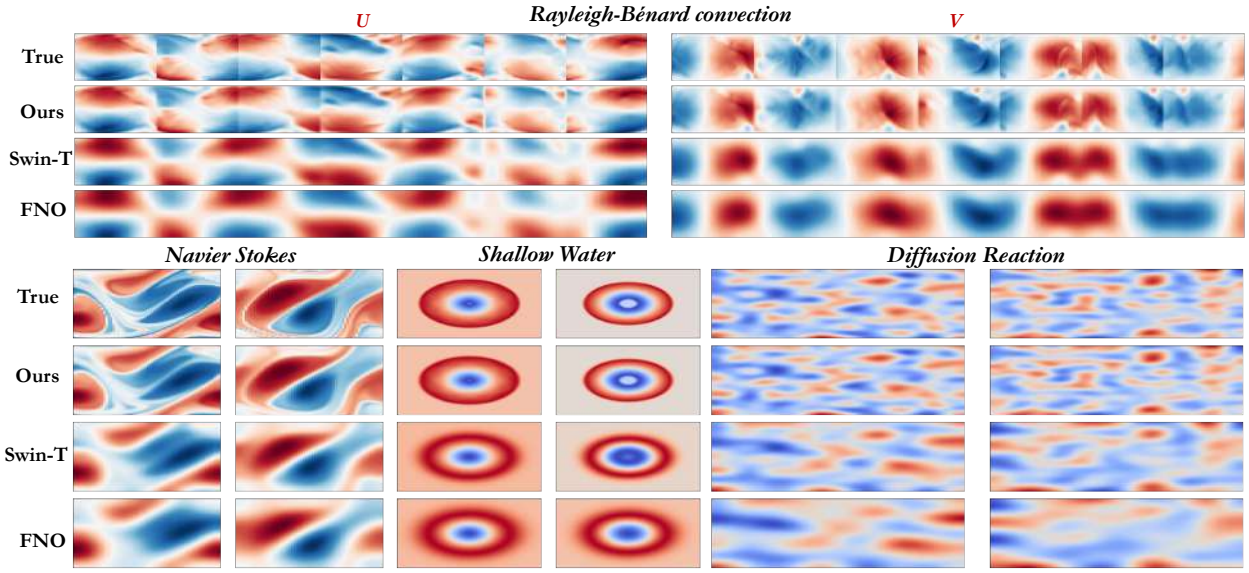

Figure 2: Our method compared to other baseline models under different physical equations. The visualization shows the true values and predicted values for Rayleigh-Bénard convection, Navier-Stokes equations, shallow water equations, and diffusion-reaction equations. All images represent the data at the final time step.

FLOPs are 686.36M and 652.40M, parameter counts are 9.68M and 50.10M, and inference times are 0.68s and 0.85s; FNO's FLOPs are 743.15M and 591.25M, parameter counts are 38.71M and 40.38M, and inference times are 0.87s and 0.83s.

Looking at the mean squared error (MSE) and structural similarity index (SSIM), ManiEarth also outperforms the other models. In terms of MSE, ManiEarth achieves 30.05 and 21.15, while ResNet scores 30.32 and 25.16, and FNO scores 30.11 and 22.64. For SSIM, ManiEarth reaches 0.99 and 0.97, significantly better than ResNet's 0.98 and 0.95 and FNO's 0.98 and 0.96.

These results indicate that ManiEarth not only leads in computational efficiency but also excels in prediction accuracy and image quality. Here is the Observations:

Obs 2. Computational Efficiency: ManiEarth significantly reduces FLOPs and parameter count, which means the model requires fewer computing resources and less memory for inference. For example, compared to FNO, ManiEarth reduces FLOPs by about 23% to 61%, greatly enhancing computational efficiency. This reduction in computational load makes the model more feasible for large-scale and real-time applications.

Obs 3. Inference Time: ManiEarth excels in inference time, a critical factor for real-time applications such as weather forecasting and disaster alert systems. The inference time of ManiEarth is noticeably shorter than that of FNO and ResNet, especially on high-resolution datasets where its efficiency is more pronounced. This speed advantage ensures timely and reliable predictions, which are crucial in scenarios where rapid decision-making is necessary.

Obs 4. Prediction Accuracy: Despite significantly cutting computational costs, ManiEarth does not compromise on prediction accuracy. In many cases, it surpasses other models in terms of accuracy. This superiority is evident from comparisons of Mean Squared Error (MSE) and Structural Similarity Index Measure (SSIM). ManiEarth maintains a low error rate while ensuring high similarity between predicted results and real data. This balance of efficiency and accuracy makes it a robust choice for complex dynamic system modeling. Overall, ManiEarth demonstrates significant advantages in physical dynamics modeling, improving computational efficiency while maintaining high prediction accuracy and image quality.

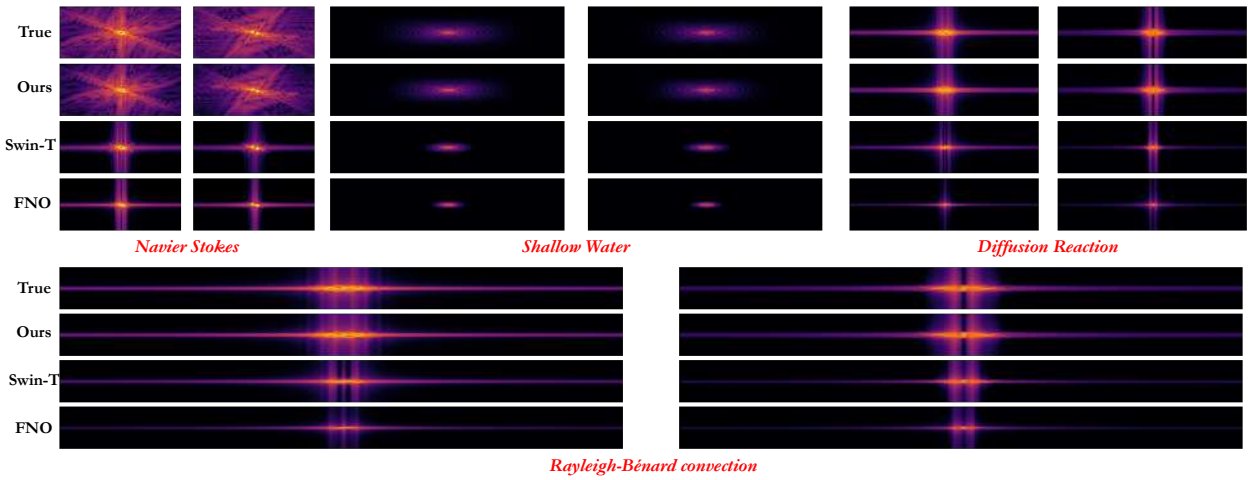

Figure 3: **Energy Spectrum Comparison Results:** This showcases the energy spectrum comparison between our method (Ours) and other benchmark models (Swin-Transformer and FNO) under four typical physical equations (Navier-Stokes equation, shallow water wave equation, diffusion-reaction equation, and Rayleigh-Bénard convection). The results show that our method accurately captures the true energy distribution of physical systems in all test scenarios, demonstrating excellent physical consistency.

4.5. ManiEarth can effectively capture physical consistency (RQ3)

The energy spectrum is an important metric for evaluating physical dynamic system models. It analyzes the energy distribution in the system and verifies the model's physical consistency. As shown in Figure 3, here are the energy spectrum comparison results between our method and other benchmark models under four typical physical equations:

Obs 5. ManiEarth can effectively capture physical consistency: The Navier-Stokes equation describes the basic principles of fluid dynamics, and its energy spectrum reflects the energy distribution in fluid motion. Our method's (Ours) energy spectrum is highly consistent with the true system (True), accurately capturing the fluid's dynamic changes and demonstrating excellent physical consistency. In contrast, Swin-Transformer (Swin-T) and Fourier Neural Operator (FNO) show significant deviations in high and low-frequency energy distributions, failing to adequately reflect the true physical dynamics of the Navier-Stokes system. In the shallow water wave equation, our method's energy spectrum closely matches the true system, indicating that the model effectively learns the physical consistency of shallow water waves. Swin-T and FNO have significant gaps in energy distribution compared to the true values, particularly in the high-frequency energy part, where they perform inadequately. The diffusion-reaction equation simulates material diffusion and chemical reaction processes. Our method excels in capturing the energy distribution, closely matching the true values, proving the model's understanding of the physical consistency of the diffusion-reaction process. Swin-T and FNO's energy spectra poorly match the true values, failing to adequately reflect the system's physical characteristics. Rayleigh-Bénard convection describes convection phenomena caused by bottom heating. Our method's energy spectrum is highly consistent with the true values, accurately capturing the energy changes in the convection process. Swin-T and FNO show deviations in the energy spectrum, unable to comprehensively reflect the system's physical dynamics.

Overall, our method significantly outperforms other benchmark models in energy spectrum performance. Whether in the Navier-Stokes equation, shallow water wave equation, diffusion-reaction equation, or Rayleigh-Bénard convection, our method accurately captures the true energy distribution of physical systems, demonstrating excellent physical consistency.

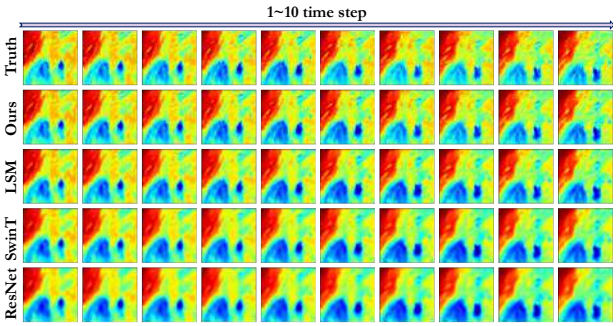

Figure 4: SEVIR visualization results mainly analyze extreme events. We see that the method proposed in this paper aligns closely with real-world conditions.

4.6. Extreme weather forecasting of ManiEarth (RQ4)

Extreme weather forecasting has always been regarded as a highly challenging task with significant real-world implications [23, 7]. SEVIR contains a large number of high-quality extreme weather events, such as storms and hurricanes, so we conduct the following experiments:

Obs 6. ManiEarth excels in predicting extreme events: Figure 4 shows our method's excellent performance in capturing extreme events. Compared to other benchmark models (LSM, SwinT, and ResNet), our method demonstrates highly consistent results with real data in SEVIR predictions from the 1st to the 10th time step. Specifically, each row in the figure shows the true data and the predictions from different models in sequence. Our method (second row) accurately captures the trend and details of extreme events, while other benchmark models show significant errors in multiple time steps. This precise capture capability indicates that our method excels not only in regular predictions but also in handling complex and extreme events. This further validates the effectiveness of our method in physical consistency and multimodal data integration, providing strong support for accurate predictions of extreme events.

5. Conclusion

In this study, we propose a novel framework that combines large language models (LLMs) with manifold learning for physical dynamics modeling. Our method significantly outperforms existing baseline models across multiple benchmark datasets. For the unknown equation SEVIR dataset, our method achieves an error of 0.1608, much lower than U-Net's 2.0280 and ResNet's 2.0787. Additionally, on the Kuroshio dataset, our method records an error of 0.0399, also outperforming other baselines. In classic physical scenarios with known equations, such as Navier-Stokes, Shallow Water, and Rayleigh-Bénard Convection, our method achieves errors of 0.2465, 0.0025, and 0.1392 respectively, all significantly better than other models. Our approach not only improves prediction accuracy but also excels in computational efficiency and physical consistency. By integrating LLMs for multimodal data and using manifold learning for efficient dimensionality reduction, our model enhances training and inference speed while improving adaptability and prediction accuracy for complex physical systems. In summary, our framework shows strong potential in physical dynamics modeling, providing an effective solution for current challenges in data utilization and computational efficiency.

Data Availability

The datasets trained during and/or analyzed during the current study are available in the original references. At the same time, you can also find it at the link: Google Drive Folder.

References

[1] Y. Wang, M. Long, J. Wang, Z. Gao, P. S. Yu, Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms, Advances in neural information processing systems 30 (2017).

[2] Z. Gao, C. Tan, L. Wu, S. Z. Li, Simvp: Simpler yet better video prediction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 3170–3180.

[3] H. Wu, Y. Liang, W. Xiong, Z. Zhou, W. Huang, S. Wang, K. Wang, Earthfarsser: Versatile spatio-temporal dynamical systems modeling in one model, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 15906–15914.

[4] K. Wang, H. Wu, G. Zhang, J. Fang, Y. Liang, Y. Wu, R. Zimmermann, Y. Wang, Modeling spatio-temporal

dynamical systems with neural discrete learning and levels-of-experts, IEEE Transactions on Knowledge and Data Engineering (2024).

[5] H. Wu, T. Hu, H. Luo, J. Wang, M. Long, Solving high-dimensional pdes with latent spectral models, arXiv preprint arXiv:2301.12664 (2023).

[6] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, arXiv preprint arXiv:2010.08895 (2020).

[7] K. Wang, H. Wu, Y. Duan, G. Zhang, K. Wang, X. Peng, Y. Zheng, Y. Liang, Y. Wang, Nuwadynamics: Discovering and updating in causal spatio-temporal modeling, in: The Twelfth International Conference on Learning Representations, 2024.

[8] G. Jin, Y. Liang, Y. Fang, Z. Shao, J. Huang, J. Zhang, Y. Zheng, Spatio-temporal graph neural networks for predictive learning in urban computing: A survey, IEEE Transactions on Knowledge and Data Engineering (2023).

[9] Z. Pan, W. Zhang, Y. Liang, W. Zhang, Y. Yu, J. Zhang, Y. Zheng, Spatio-temporal meta learning for urban traffic prediction, IEEE Transactions on Knowledge and Data Engineering 34 (3) (2020) 1462–1476.

[10] Y. Liang, Y. Xia, S. Ke, Y. Wang, Q. Wen, J. Zhang, Y. Zheng, R. Zimmermann, Airformer: Predicting nationwide air quality in china with transformers, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 14329–14337.

[11] X. Jiang, J. Ma, G. Xiao, Z. Shao, X. Guo, A review of multimodal image matching: Methods and applications, Information Fusion 73 (2021) 22–71.

[12] H. Wang, T. Fu, Y. Du, W. Gao, K. Huang, Z. Liu, P. Chandak, S. Liu, P. Van Katwyk, A. Deac, et al., Scientific discovery in the age of artificial intelligence, Nature 620 (7972) (2023) 47–60.

[13] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nature Reviews Physics 3 (6) (2021) 422–440.

[14] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-c. Woo, Convolutional lstm network: A machine learning approach for precipitation nowcasting, Advances in neural information processing systems 28 (2015).

[15] B. Raonic, R. Molinaro, T. Rohner, S. Mishra, E. de Bezenac, Convolutional neural operators, in: ICLR 2023 Workshop on Physics for Machine Learning, 2023.

[16] J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, T. Kurth, D. Hall, Z. Li, K. Azizzadenesheli, et al., Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators, arXiv preprint arXiv:2202.11214 (2022).

[17] M. Kiani Shahvandi, M. Schartner, B. Soja, Neural ode differential learning and its application in polar motion prediction, Journal of Geophysical Research: Solid Earth 127 (11) (2022) e2022JB024775.

[18] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: Learning maps between function spaces, arXiv preprint arXiv:2108.08481 (2021).

[19] N. B. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. M. Stuart, A. Anandkumar, Neural operator: Learning maps between function spaces with applications to pdes., J. Mach. Learn. Res. 24 (89) (2023) 1–97.

[20] H. Wu, S. Zhou, X. Huang, W. Xiong, Neural manifold operators for learning the evolution of physical dynamics (2024).
URL https://openreview.net/forum?id=SQnOmOzqAM

[21] S. Cai, Z. Mao, Z. Wang, M. Yin, G. E. Karniadakis, Physics-informed neural networks (pinns) for fluid mechanics: A review, Acta Mechanica Sinica 37 (12) (2021) 1727–1738.

[22] Z. Li, H. Zheng, N. Kovachki, D. Jin, H. Chen, B. Liu, K. Azizzadenesheli, A. Anandkumar, Physics-informed neural operator for learning partial differential equations, ACM/JMS Journal of Data Science 1 (3) (2024) 1–27.

[23] H. Wu, W. Xiong, F. Xu, X. Luo, C. Chen, X.-S. Hua, H. Wang, Pastnet: Introducing physical inductive biases for spatio-temporal video prediction, arXiv preprint arXiv:2305.11421 (2023).

[24] P. Rajendra, V. Brahmajirao, Modeling of dynamical systems through deep learning, Biophysical Reviews 12 (6) (2020) 1311–1320.

[25] B. Lusch, J. N. Kutz, S. L. Brunton, Deep learning for universal linear embeddings of nonlinear dynamics, Nature communications 9 (1) (2018) 4950.

[26] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, Q. Tian, Accurate medium-range global weather forecasting with 3d neural networks, Nature 619 (7970) (2023) 533–538.

[27] Y. Zhang, M. Long, K. Chen, L. Xing, R. Jin, M. I. Jordan, J. Wang, Skilful nowcasting of extreme precipitation with nowcastnet, Nature 619 (7970) (2023) 526–532.

[28] H. Wang, J. Li, A. Dwivedi, K. Hara, T. Wu, Beno: Boundary-embedded neural operators for elliptic pdes, arXiv preprint arXiv:2401.09323 (2024).

[29] Y. Wang, H. Wu, J. Zhang, Z. Gao, J. Wang, S. Y. Philip, M. Long, Predrnn: A recurrent neural network for spatiotemporal predictive learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (2) (2022) 2208–2225.

[30] C. Tan, Z. Gao, S. Li, S. Z. Li, Simvp: Towards simple yet powerful spatiotemporal predictive learning, arXiv preprint arXiv:2211.12509 (2022).

[31] J. Xu, X. Liu, Y. Wu, Y. Tong, Q. Li, M. Ding, J. Tang, Y. Dong, Imagereward: Learning and evaluating human preferences for text-to-image generation, Advances in Neural Information Processing Systems 36 (2024).

[32] A. Raj, S. Kaza, B. Poole, M. Niemeyer, N. Ruiz, B. Mildenhall, S. Zada, K. Aberman, M. Rubinstein, J. Barron, et al., Dreambooth3d: Subject-driven text-to-3d generation, in: Proceedings of the IEEE/CVF international conference on computer vision, 2023, pp. 2349–2359.

[33] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., Gpt-4 technical report, arXiv preprint arXiv:2303.08774 (2023).

[34] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[35] D. Lahat, T. Adali, C. Jutten, Multimodal data fusion: an overview of methods, challenges, and prospects, Proceedings of the IEEE 103 (9) (2015) 1449–1477.

[36] R. Bramon, I. Boada, A. Bardera, J. Rodriguez, M. Feixas, J. Puig, M. Sbert, Multimodal data fusion based on mutual information, IEEE Transactions on Visualization and Computer Graphics 18 (9) (2011) 1574–1587.

[37] M. Moayeri, K. Rezaei, M. Sanjabi, S. Feizi, Text-to-concept (and back) via cross-model alignment, in: International Conference on Machine Learning, PMLR, 2023, pp. 25037–25060.

[38] S. Wu, H. Fei, L. Qu, W. Ji, T.-S. Chua, Next-gpt: Any-to-any multimodal llm, arXiv preprint arXiv:2309.05519 (2023).

[39] D. Zhang, Y. Yu, C. Li, J. Dong, D. Su, C. Chu, D. Yu, Mm-llms: Recent advances in multimodal large language models, arXiv preprint arXiv:2401.13601 (2024).

[40] N. Bai, R. da Silva Torres, A. Fensel, T. Metze, A. Dewulf, Inferring climate change stances from multimodal tweets, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 2467–2471.

[41] R. Chandra, S. S. Kumar, R. Patra, S. Agarwal, Decision support system for forest fire management using ontology with big data and llms, arXiv preprint arXiv:2405.11346 (2024).

[42] X. Wang, M. Fang, Z. Zeng, T. Cheng, Where would i go next? large language models as human mobility predictors, arXiv preprint arXiv:2308.15197 (2023).

[43] K. Fukami, K. Taira, Grasping extreme aerodynamics on a low-dimensional manifold, Nature Communications 14 (1) (2023) 6480.

[44] B. Chen, K. Huang, S. Raghupathi, I. Chandratreya, Q. Du, H. Lipson, Automated discovery of fundamental variables hidden in experimental data, Nature Computational Science 2 (7) (2022) 433–442.

[45] R. Talmon, S. Mallat, H. Zaveri, R. R. Coifman, Manifold learning for latent variable inference in dynamical systems, IEEE Transactions on Signal Processing 63 (15) (2015) 3843–3856.

[46] T. Shnitzer, R. Talmon, J.-J. Slotine, Manifold learning for data-driven dynamical system analysis, The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications (2020) 359–382.

[47] M. Han, S. Feng, C. P. Chen, M. Xu, T. Qiu, Structured manifold broad learning system: A manifold perspective for large-scale chaotic time series analysis and prediction, IEEE Transactions on Knowledge and Data Engineering 31 (9) (2018) 1809–1821.

[48] Y. Ge, W. Hua, K. Mei, J. Tan, S. Xu, Z. Li, Y. Zhang, et al., Openagi: When llm meets domain experts, Advances in Neural Information Processing Systems 36 (2024).

[49] Z. Hang, Y. Ma, H. Wu, H. Wang, M. Long, Unisolver: Pde-conditional transformers are universal pde solvers, arXiv preprint arXiv:2405.17527 (2024).

[50] N. B. Erichson, M. Muehlebach, M. W. Mahoney, Physics-informed autoencoders for lyapunov-stable fluid flow prediction, arXiv preprint arXiv:1905.10866 (2019).

[51] R. Wang, K. Kashinath, M. Mustafa, A. Albert, R. Yu, Towards physics-informed deep learning for turbulent flow prediction, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 1457–1466.

[52] Q. Zhu, Z. Liu, J. Yan, Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks, Computational Mechanics 67 (2021) 619–635.

[53] P. Shokouhi, V. Kumar, S. Prathipati, S. A. Hosseini, C. L. Giles, D. Kifer, Physics-informed deep learning for prediction of co2 storage site response, Journal of Contaminant Hydrology 241 (2021) 103835.

[54] J.-X. Wang, J. Huang, L. Duan, H. Xiao, Prediction of reynolds stresses in high-mach-number turbulent boundary layers using physics-informed machine learning, Theoretical and Computational Fluid Dynamics 33 (2019) 1–19.

[55] S. Wang, P. Perdikaris, Long-time integration of parametric evolution equations with physics-informed deeponets, Journal of Computational Physics 475 (2023) 111855.

[56] F. M. Rohrhofer, S. Posch, C. Gößnitzer, B. C. Geiger, Understanding the difficulty of training physics-informed neural networks on dynamical systems, arXiv preprint arXiv:2203.13648 (2022).

[57] S. Wang, X. Yu, P. Perdikaris, When and why pinns fail to train: A neural tangent kernel perspective, Journal of Computational Physics 449 (2022) 110768.

[58] Z. Long, Y. Lu, X. Ma, B. Dong, Pde-net: Learning pdes from data, in: International conference on machine learning, PMLR, 2018, pp. 3208–3216.

[59] R. T. Chen, Y. Rubanova, J. Bettencourt, D. K. Duvenaud, Neural ordinary differential equations, Advances in neural information processing systems 31 (2018).

[60] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[61] M. Höge, A. Scheidegger, M. Baity-Jesi, C. Albert, F. Fenicia, Improving hydrologic models for predictions and process understanding using neural odes, Hydrology and Earth System Sciences 26 (19) (2022) 5085–5102.

[62] V. Mehta, I. Char, W. Neiswanger, Y. Chung, A. Nelson, M. Boyer, E. Kolemen, J. Schneider, Neural dynamical systems: Balancing structure and flexibility in physical prediction, in: 2021 60th IEEE Conference on Decision and Control (CDC), IEEE, 2021, pp. 3735–3742.

[63] J. E. Gerken, J. Aronsson, O. Carlsson, H. Linander, F. Ohlsson, C. Petersson, D. Persson, Geometric deep learning and equivariant neural networks, arXiv preprint arXiv:2105.13926 (2021).

[64] N. Dehmamy, R. Walters, Y. Liu, D. Wang, R. Yu, Automatic symmetry discovery with lie algebra convolutional network, Advances in Neural Information Processing Systems 34 (2021) 2503–2515.

[65] S. Villar, D. W. Hogg, K. Storey-Fisher, W. Yao, B. Blum-Smith, Scalars are universal: Equivariant machine learning, structured like classical physics, Advances in Neural Information Processing Systems 34 (2021) 28848–28863.

[66] J. Brandstetter, M. Welling, D. E. Worrall, Lie point symmetry data augmentation for neural pde solvers, in: International Conference on Machine Learning, PMLR, 2022, pp. 2241–2256.

[67] R. Walters, J. Li, R. Yu, Trajectory prediction using equivariant continuous convolution, arXiv preprint arXiv:2010.11344 (2020).

[68] R. Wang, R. Walters, R. Yu, Approximately equivariant networks for imperfectly symmetric dynamics, in: International Conference on Machine Learning, PMLR, 2022, pp. 23078–23091.

[69] G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of control, signals and systems 2 (4) (1989) 303–314.

[70] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural networks 2 (5) (1989) 359–366.

[71] L. Lu, P. Jin, G. Pang, Z. Zhang, G. E. Karniadakis, Learning nonlinear operators via deeponet based on the universal approximation theorem of operators, Nature machine intelligence 3 (3) (2021) 218–229.

[72] B. O. Koopman, Hamiltonian systems and transformation in hilbert space, Proceedings of the National Academy of Sciences 17 (5) (1931) 315–318.

[73] E. Yeung, S. Kundu, N. Hodas, Learning deep neural network representations for koopman operators of nonlinear dynamical systems, in: 2019 American Control Conference (ACC), IEEE, 2019, pp. 4832–4839.

[74] W. Xiong, X. Huang, Z. Zhang, R. Deng, P. Sun, Y. Tian, Koopman neural operator as a mesh-free solver of non-linear partial differential equations, arXiv preprint arXiv:2301.10022 (2023).

[75] W. Xiong, M. Ma, X. Huang, Z. Zhang, P. Sun, Y. Tian, Koopmanlab: machine learning for solving complex physics equations, APL Machine Learning 1 (3) (2023).

[76] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: Graph kernel network for partial differential equations, arXiv preprint arXiv:2003.03485 (2020).

[77] Z. Li, H. Zheng, N. Kovachki, D. Jin, H. Chen, B. Liu, K. Azizzadenesheli, A. Anandkumar, Physics-informed neural operator for learning partial differential equations, arXiv preprint arXiv:2111.03794 (2021).

[78] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, A. Stuart, K. Bhattacharya, A. Anandkumar, Multipole graph neural operator for parametric partial differential equations, Advances in Neural Information Processing Systems 33 (2020) 6755–6766.

[79] T. Tripura, S. Chakraborty, Wavelet neural operator: a neural operator for parametric partial differential equations, arXiv preprint arXiv:2205.02191 (2022).

[80] K. Champion, B. Lusch, J. N. Kutz, S. L. Brunton, Data-driven discovery of coordinates and governing equations, Proceedings of the National Academy of Sciences 116 (45) (2019) 22445–22451.

[81] D. Floryan, M. D. Graham, Data-driven discovery of intrinsic dynamics, Nature Machine Intelligence (2022) 1–8.

[82] Z. Bai, P. M. Dewilde, R. W. Freund, Reduced-order modeling, Handbook of numerical analysis 13 (2005) 825–895.

[83] K. Lee, K. T. Carlberg, Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders, Journal of Computational Physics 404 (2020) 108973.

[84] S. Fresca, L. Dede', A. Manzoni, A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized pdes, Journal of Scientific Computing 87 (2021) 1–36.

[85] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, Journal of fluid mechanics 656 (2010) 5–28.

[86] P. J. Schmid, Dynamic mode decomposition and its variants, Annual Review of Fluid Mechanics 54 (2022) 225–254.

[87] J. Seidman, G. Kissas, P. Perdikaris, G. J. Pappas, Nomad: Nonlinear manifold decoders for operator learning, Advances in Neural Information Processing Systems 35 (2022) 5601–5613.

[88] S. Lanthaler, Operator learning with pca-net: upper and lower complexity bounds, arXiv preprint arXiv:2303.16317 (2023).

[89] O. Ronneberger, P. Fischer, T. 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, Springer, 2015, pp. 234–241.

[90] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 10012–10022.

[91] A. Tran, A. Mathews, L. Xie, C. S. Ong, Factorized fourier neural operators, arXiv preprint arXiv:2111.13802 (2021).