PURE: Prompt EvolUtion with GRaph ODE for Out-of-distribution Fluid Dynamics Modeling

Hao Wu[1,3], Changhu Wang[2], Fan Xu[1], Jinbao Xue[3], Chong Chen[4], Xian-Sheng Hua[4], Xiao Luo[2,*]

[1]*University of Science and Technology of China,* [2]*University of California, Los Angeles,* [3]*Tencent ,* [4]*Terminus Group*

TL;DR

This paper introduces PURE, a method for out-of-distribution fluid dynamics modeling. PURE uses a graph ODE to learn time-evolving prompts, adapting models to distribution shifts from system changes and temporal evolution. It enhances robustness by minimizing mutual information between prompts and observations. Experiments confirm PURE's superiority over baselines.

Problem Definition

The problem involves predicting future sensor observations in a fluid dynamical system with N sensors. Given historical observations $\boldsymbol{s}_i^{1:T_0}$ at sensor locations $\boldsymbol{x}_i$, and system parameters $\boldsymbol{\xi}$ (e.g., PDE coefficients), the goal is to forecast future observations $\boldsymbol{s}_i^{T_0+1:T_0+T}$. The challenge lies in handling distribution shifts caused by variations in system parameters and time, where training and test distributions differ. This is formulated as learning a function f that maps historical data $\boldsymbol{u}^{1:T_0}$ to future observations $\boldsymbol{u}^{T_0+1:T_0+T}$.

Three Contributions

(1) ***Problem Connection.*** *We are the first to connect prompt learning with dynamical system modeling to solve the issue of out-of-distribution shifts.*

(2) ***Novel Methodology.*** *Our PURE first learns from historical observations and system parameters to initialize prompt embeddings and then adopts a graph ODE with the interpolation of observation sequences to capture their continuous evolution for model adaptation under out-of-distribution shifts.*

(3) ***Superior Performance.*** *Comprehensive experiments validate the effectiveness of our PURE in different challenging settings.*

Our Approach

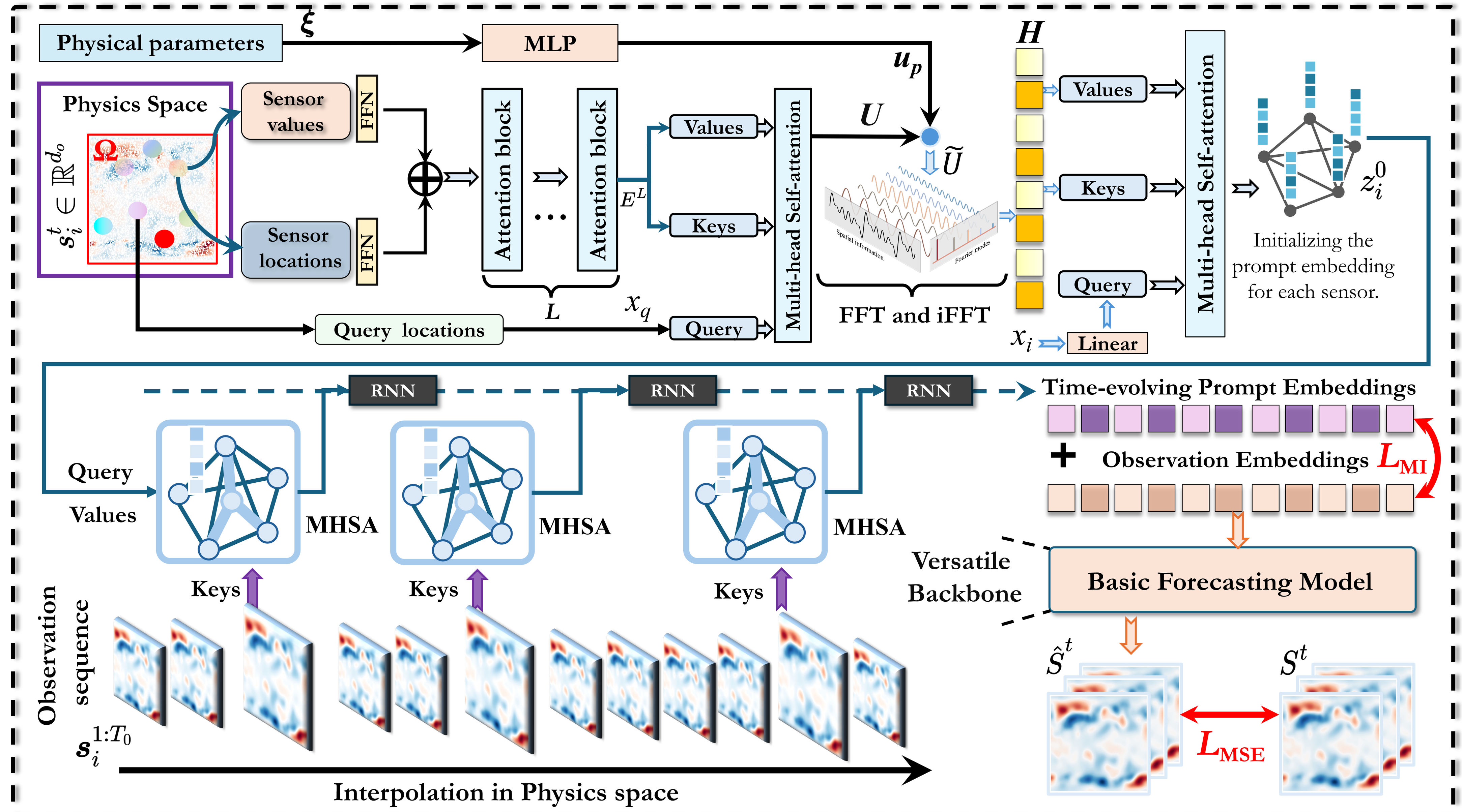

Figure 1. Overview of the PURE framework

1. Observation and Prediction

The goal is to predict future observations $\boldsymbol{u}^{T_0+1:T_0+T}$ from historical data $\boldsymbol{u}^{1:T_0}$, expressed as:

$$\boldsymbol{u}^{T_0+1:T_0+T} = f(\boldsymbol{u}^{1:T_0}). \tag{1}$$

2. Embedding-based Prediction

To address distribution shifts, we introduce invariant observation embeddings $\boldsymbol{\mu}^t$ and prompt embeddings $\boldsymbol{z}^t$, which are used to generate the final prediction:

$$u_{\text{output}} = \phi([\boldsymbol{\mu}^t, \boldsymbol{z}^t]). \tag{2}$$

Here, $\boldsymbol{z}^t \perp \boldsymbol{\mu}^t$, indicating that prompt and observation embeddings are independent.

3. Time-evolving Prompt Learning

The time evolution of prompt embeddings is modeled using a Graph ODE, described as:

$$\frac{d\boldsymbol{z}_i^t}{dt} = \psi_a\left(\sum_{j\in\mathcal{S}^t(i)} \text{softmax}\left(\frac{[\tilde{\boldsymbol{W}}^Q\boldsymbol{z}_i^t]^T\cdot[\tilde{\boldsymbol{W}}^K\boldsymbol{s}_j^t]}{\sqrt{d}}\right)\cdot\psi_r([\boldsymbol{z}_i^t,\boldsymbol{z}_j^t])\right), \tag{3}$$

where $\mathcal{S}^t(i)$ represents the set of neighboring sensors at time t.

4. Loss Function The optimization objective includes both the mean squared error (MSE) and mutual information minimization (MI) losses, combined as:

$$\mathcal{L} = \mathcal{L}_{MSE} + \lambda\mathcal{L}_{MI}. \tag{4}$$

Results

Table 1: We compare our study's performance with 10 baselines.

Model	Prometheus w/o OOD	Prometheus w/ OOD	Navier–Stokes w/o OOD	Navier–Stokes w/ OOD	Spherical-SWE w/o OOD	Spherical-SWE w/ OOD	3D Reaction–Diff w/o OOD	3D Reaction–Diff w/ OOD	ERA5 w/o OOD	ERA5 w/ OOD
U-Net [59]	0.0931	**0.1067**	0.1982	0.2243	0.0083	0.0087	0.0148	0.0183	0.0843	0.0932
ResNet [20]	0.0674	0.0696	0.1823	0.2301	0.0081	0.0192	0.0151	0.0186	0.0921	0.0977
ViT [9]	0.0632	0.0691	**0.2342**	**0.2621**	0.0065	0.0072	0.0157	0.0192	0.0762	0.0786
SwinT [47]	0.0652	0.0729	0.2248	0.2554	0.0062	0.0068	0.0155	0.0190	0.0782	0.0832
FNO [43]	**0.0447**	**0.0506**	0.1556	0.1712	0.0038	0.0045	0.0132	0.0179	0.7233	**0.9821**
UNO [1]	0.0532	0.0643	0.1764	0.1984	0.0034	0.0041	**0.0121**	0.0164	0.6652	0.7621
CNO [58]	0.0542	0.0655	0.1473	0.1522	0.0037	0.0038	0.0145	0.0182	0.5243	0.7821
NMO [73]	0.0397	0.0483	**0.1021**	**0.1032**	**0.0026**	0.0031	0.0129	**0.0168**	**0.0432**	**0.0563**
CGODE [25]	0.0761	0.0843	0.2035	0.2243	**0.0873**	**0.0987**	**0.8371**	**0.9261**	**0.8721**	0.9872
DGODE [72]	**0.0344**	**0.0359**	**0.0805**	0.0925	**0.0022**	**0.0028**	0.0122	0.0156	0.0543	0.0635
Ours + PURE	**0.0323**	**0.0328**	**0.0752**	**0.0763**	**0.0022**	**0.0024**	**0.0119**	**0.0127**	**0.0398**	**0.0401**
Promotion	6.10%	8.63%	6.58%	26.07%	0.00%	16.67%	1.65%	22.56%	7.87%	28.77%

Table 2: This table shows the performance of the PURE framework.

Model	Prometheus Ori	Prometheus +PURE	Navier–Stokes Ori	Navier–Stokes +PURE	Spherical-SWE Ori	Spherical-SWE +PURE	3D Reaction–Diff Ori	3D Reaction–Diff +PURE	ERA5 Ori	ERA5 +PURE
ResNet [20]	0.0674	0.0542	0.1823	0.1492	0.0081	0.0067	0.0151	0.0141	0.0921	0.0896
NMO [9]	0.0397	0.0281	0.1021	0.0876	0.0026	0.0012	0.0129	0.0123	0.0432	0.0389
DGODE [47]	0.0344	0.0201	0.0805	0.0792	0.0022	0.0020	0.0122	0.0110	0.0543	0.0462

Table 3: Comparison of Spatial & Temporal Generalization.

Sparsity / Train ↓	Test→	$s_{TS}=5\%$ In-T	$s_{TS}=5\%$ Out-T	$s_{TS}=25\%$ In-T	$s_{TS}=25\%$ Out-T	$s_{TS}=50\%$ In-T	$s_{TS}=50\%$ Out-T	$s_{TS}=75\%$ In-T	$s_{TS}=75\%$ Out-T
$s_{TR}=75\%$	U-Net	0.1847	0.2103	0.2345	0.2877	0.2654	0.3018	0.2273	0.3391
	+ PURE	0.1622	0.1854	0.2079	0.2581	0.2365	0.2710	0.1998	0.3024
	FNO	0.0659	0.0872	0.0921	0.1232	0.1109	0.1821	0.2109	0.2455
	+ PURE	0.0504	0.0654	0.0689	0.0946	0.0805	0.1417	0.1582	0.1883

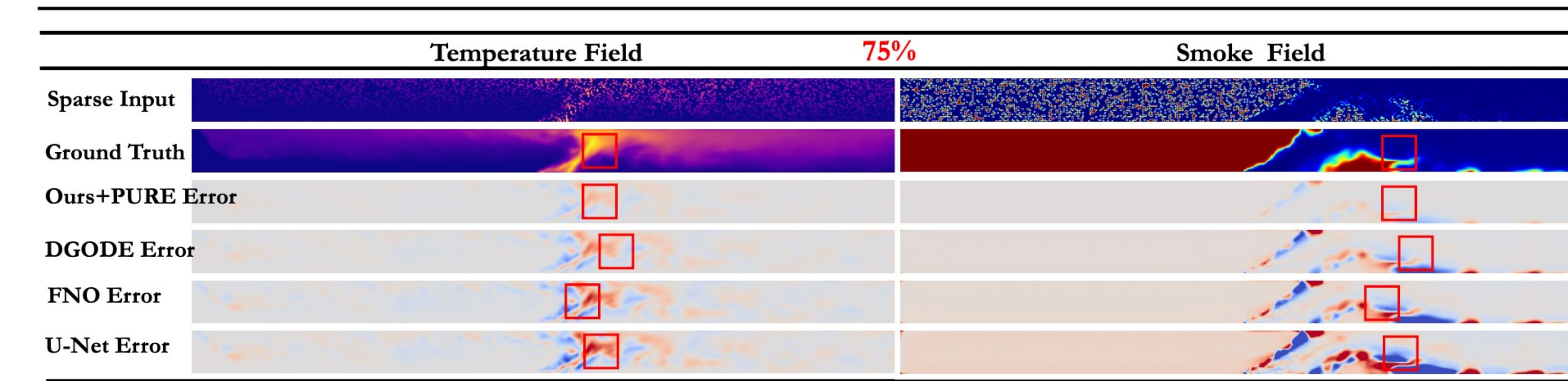

Figure 2: The sparse input data used for predictions.

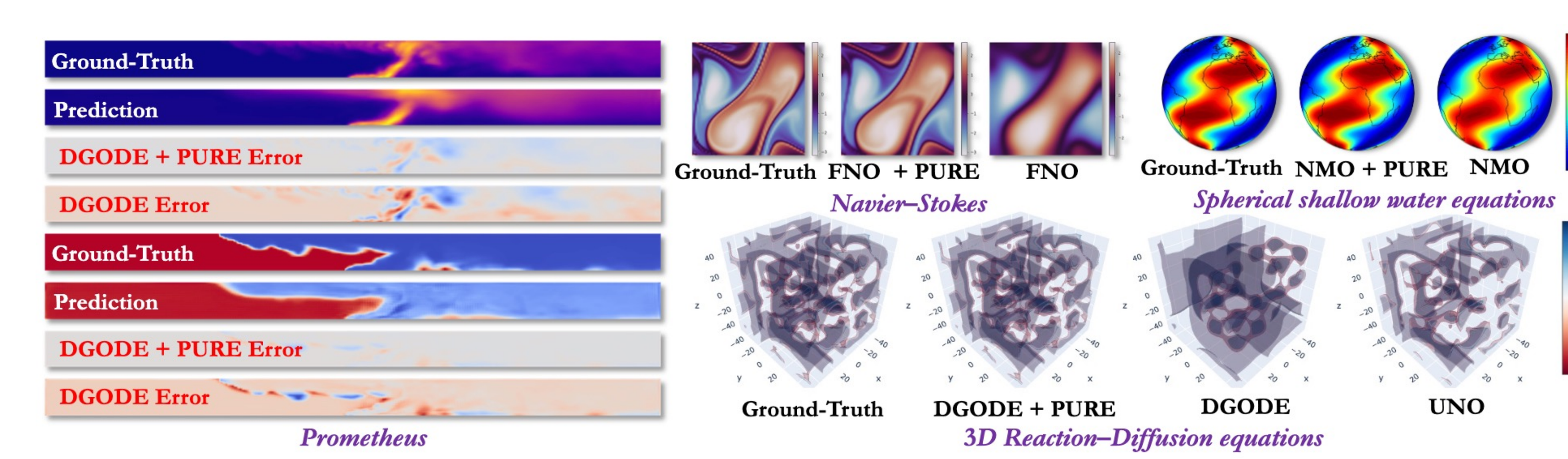

Figure 3: The Figure compares the performance of various methods in fluid dynamics modeling.

For more information, please refer to out full paper published in NeurIPS 2024!