How to Re-enable PDE Loss for Physical Systems Modeling Under Partial Observation

Haodong Feng[1,2], **Yue Wang**[3] *, **Dixia Fan**[2] *
[1] Zhejiang University
[2] School of Engineering, Westlake University
[3] Microsoft Research
{fenghaodong, fandixia}@westlake.edu.cn, yuwang5@microsoft.com

Abstract

In science and engineering, machine learning techniques are increasingly successful in physical systems modeling (predicting future states of physical systems). Effectively integrating PDE loss as a constraint of system transition can improve the model's prediction by overcoming generalization issues due to data scarcity, especially when data acquisition is costly. However, in many real-world scenarios, due to sensor limitations, the data we can obtain is often only partial observation, making the calculation of PDE loss seem to be infeasible, as the PDE loss heavily relies on high-resolution states. We carefully study this problem and propose a novel framework named **R**e-enable **P**DE **L**oss under **P**artial **O**bservation (RPLPO). The key idea is that although enabling PDE loss to constrain system transition solely is infeasible, we can re-enable PDE loss by reconstructing the learnable high-resolution state and constraining system transition simultaneously. Specifically, RPLPO combines an encoding module for reconstructing learnable high-resolution states with a transition module for predicting future states. The two modules are jointly trained by data and PDE loss. We conduct experiments in various physical systems to demonstrate that RPLPO has significant improvement in generalization, even when observation is sparse, irregular, noisy, and PDE is inaccurate.

Code — https://github.com/HDFengChina/RPLPO
Extended version — https://arxiv.org/abs/2412.09116

Introduction

Using machine learning methods to model physical systems has become a promising direction in science and engineering, e.g., fluid dynamics, diffusion, and so on (An, Kim, and James 2008; Zhang et al. 2023; Wang et al. 2023). Since the data collection procedure is always both time and cost-consuming, a limited amount of data will hurt the generalization of the physical system modeling. Recently, some works have introduced PDE loss in training models and achieved promising performance (Li et al. 2021b; Gao, Sun, and Wang 2021a; Goswami et al. 2022a; Ren et al. 2023b; Huang et al. 2023) to reduce the use of costly data and improve generalization capacity of physical system models. Most works rely on high-resolution states to calculate or approximate the derivative for PDE loss. However, due to sensor limitations (Iakovlev, Heinonen, and Lähdesmäki 2020; Lütjens et al. 2022; Yin et al. 2022; Boussif et al. 2022; Iakovlev, Heinonen, and Lähdesmäki 2023), the application of PDE loss for modeling physical systems is significantly challenged by the partially observable nature of measurement. Computing partial derivatives in PDE loss with partial observation introduces significant biases, leading to a dilemma: either we avoid using PDE loss in the model, thereby weakening its generalization, or we use the biased PDE loss, which can also hurt the model's performance. As a result, a scientific problem has been emerged:

Whether and how can we improve the model's generalization capacity for physical systems modeling by using PDE loss under partial observation?

In this work, we propose a novel framework named RPLPO to re-enable PDE loss for physical systems modeling under partial observation, thereby enhancing the model's capacity to generalize and predict future partially observable states. By using RPLPO, the PDE loss can be used to reconstruct the corresponding high-resolution state by only using partial observation and re-enable the PDE loss on the transition between adjacent learnable high-resolution states corresponding to the partial observations simultaneously.

RPLPO mainly comprises an encoding module and a transition module. The fundamental procedure is to reconstruct the learnable high-resolution state from partial observation by using the encoding module and then predict the subsequent state with the transition module. However, it is hard to train the encoding module without available high-resolution data through data-driven supervised methods. Therefore, we use several recent consecutive observations to reconstruct a learnable high-resolution state, then train it via a PDE loss.

We design to train the encoding and transition modules jointly with two periods. In the **base-training period**, the encoding module is trained using a PDE loss without requiring high-resolution data. The transition module is trained collaboratively using the data loss and PDE loss to overcome the generalization issues due to data scarcity. In the **two-stage fine-tuning period**, the framework leverages unlabeled data in a semi-supervised learning manner to further improve the model's generalization. The transition module is first fine-tuned with unlabeled data and PDE loss independently; then, their information is propagated to the encoding

*Corresponding authors.

module, which is fine-tuned using data loss calculated on the original labeled data.

We demonstrate the performance of RPLPO on five PDEs dynamics, which represent common physical systems in the real world, e.g., Burgers equation, wave equation, Navier Stokes equation, linear shallow water equation, and nonlinear shallow water equation. The results show that, with our framework, learned models have significant improvements in generalization capacity to predict future partial observed states both in single-step and multi-step. Moreover, RPLPO is validated to be: **(a)** effective in different data numbers, sparsity levels, irregular partial observations, inaccurate PDE, and noisy data; **(b)** robust in implementation and hyperparameters; **(c)** computationally efficient under the premise of generalization improvement.

Our contributions can be summarized in two parts: **(1)** We propose a novel framework called RPLPO to re-enable PDE loss for physical systems modeling under partial observation to improve the model's generalization. **(2)** We demonstrate that RPLPO leads to a significant improvement in generalization for predicting the future partial observed state, than baseline methods, across various physical systems.

Related Works

We briefly discuss the highly related works here and leave the detailed version in **Technical Appendix**.

Physics-informed machine learning: More and more physics-informed machine learning methods have been proposed to learn the solutions of PDEs (Karniadakis et al. 2021), including neural operator (Lu, Jin, and Karniadakis 2019; Li et al. 2020a,b, 2021b; Wang, Wang, and Perdikaris 2021; Gupta, Xiao, and Bogdan 2021; Goswami et al. 2022b; Boussif et al. 2022; Yin et al. 2023; Iakovlev, Heinonen, and Lähdesmäki 2023; Hansen et al. 2023; Chen et al. 2023) and physics-informed neural networks (PINNs) (Raissi, Perdikaris, and Karniadakis 2019; Yang, Meng, and Karniadakis 2021; Cai et al. 2021; Karniadakis et al. 2021). These works used PDE loss to model or solve PDEs dynamics. The key to their success is the incorporation of accurate PDE, including high-resolution states or formulas. However, they cannot be applied to learn partial observation directly, as their physics-informed training manner has significant bias when calculating by partial observation.

High-resolution state reconstruction: In computer vision (CV) and physical systems modeling, several works aim to reconstruct a high-resolution state from its partial observation, also known as super-resolution (Zhu et al. 2020; Li et al. 2021a). In CV, numerous deep learning-based models (Dong et al. 2014; Lim et al. 2017; Soh et al. 2019; Nazeri, Thasarathan, and Ebrahimi 2019; Zhao et al. 2020), and generative models (Ledig et al. 2017; Liu, Siu, and Wang 2021; Gao et al. 2023) have emerged. In physical systems, the task has attracted more and more attention (Ren et al. 2023a), which aims to use PDE loss in models (Wang et al. 2020; Esmaeilzadeh et al. 2020; Fathi et al. 2020; Jangid et al. 2022; Ren et al. 2023b; Shu, Li, and Farimani 2023) or not need for data (Gao, Sun, and Wang 2021b; Kelshaw, Rigas, and Magri 2022; Zayats et al. 2022). Most of these works rely on high-resolution states to do supervised learning; only a small part relies on PDE loss, thus resulting in relatively larger reconstruction errors without leveraging data information or hard to model the unsteady dynamics (Gao, Sun, and Wang 2021b). There is a work (Rao et al. 2023) that is most related to our work. In this work, Chengpeng Rao, *et al.* also considered the high-resolution state reconstruction and the time-series prediction problem, but unlike our work, it is similar to the neural PDE solvers like PINNs rather than operator learning. It cannot generalize to different initial conditions. We have also leveraged this work as one of our baselines.

Different from the above related works, our framework does not require high-resolution states but rather shows that by a carefully designed framework, the physics-informed training manner can be incorporated with partial observation and further improve model's generalization capacity.

Problem Description

Problem setting: We aim to use PDE loss to improve model's generalization for predicting future partially observed states under partial observation in physical systems. Denote $u_t \triangleq u(t) \in \mathcal{U}$ as an observed state, we want to infer $u_{t+\tau}$ for $\tau > 0$ at any time t. $\mathcal{U}$ is the functional space of form $\Omega \to \mathbb{R}^n$, where $\Omega \subset \mathbb{R}^p$ is the set of observational point location, and n is the number of system variable. That is to say, u_t is a function of $x \in \Omega$, with vectorical output $u_t(x) \in \mathbb{R}^n$; cf. examples of Section **Experiments Setup**. In such problems, trajectories share the same dynamics but vary by their initial conditions (ICs) $u_0 \in \mathcal{U}$. We observe a finite training set of trajectories $\mathcal{D}$ with label and $\mathcal{B}$ lacks future observations (without label), using a partial spatial observation grid $\mathcal{X} \subset \Omega$ on discrete times $t \in \mathcal{T} \subset [0, T]$. In inference, the partial observation dataset is observed with $\mathcal{X}$. Note that inference is performed on test data observed from trajectories given different ICs to verify model's generalization to different trajectories.

Evaluation scenarios: We select four criteria that our framework should meet. First, models trained by RPLPO should be generalized to the change of trajectories to predict future partial observations. It is measured by relative error $\frac{\|\hat{u}_{t+\tau} - u_{t+\tau}\|_2}{\|u_{t+\tau}\|_2}$ as Eqn. 5. Second, it should also reconstruct accurate high-resolution states. It is measured by relative reconstruction error $\epsilon = \frac{\|\hat{h}_t - h_t\|_2}{\|h_t\|_2}$, where $\hat{h}_t, h_t$ are reconstruction and label of high-resolution states (only available to calculate ϵ in inference). Third, it should be generalized to multi-step prediction, which requires better generalization to reduce growth of errors. Finally, it should be effective for different data numbers, irregular observation, inaccurate PDE, and noisy data, robust to different encoding modules and parameters, and computationally efficient.

Methodology

Here, we describe the overview of our framework, its key components, and our designed training strategy.

Overview of RPLPO

As we mentioned in the introduction, the using of PDE loss in the model is crucial for the model generalization but is

significantly challenged by partial observation. To re-enable the use of PDE loss, RPLPO constructs PDE loss on the adjacent learnable high-resolution outputs of the encoding module. Then, the PDE loss can also facilitate the learning of transition process in the learnable high-resolution space. RPLPO jointly trains these two modules to effectively employ PDE loss to enhance the model's generalization. As shown Fig. 1 (left), there are two training periods: **Base-training period:** The encoding module is trained with a PDE loss without high-resolution states, while the transition module is trained collaboratively using data loss and PDE loss. **Two-stage fine-tuning period:** It utilizes unlabeled data semi-supervisedly for further improvement. The first stage involves fine-tuning (FT) transition module independently, with PDE loss calculated on unlabeled data. Then, their information is propagated to encoding module in the second stage by fine-tuning with data loss calculated on original labeled data.

Model Components

We presented all the model components in Fig. 1. Here we introduce each of them separately.

Encoding module: $E_\theta(u_t^n)$. The encoding module computes the learnable high-resolution state h_t given the partial observation. To input more temporal information, we use n temporal features of $u_t^n = \{u_{t-i*\tau}\}_{i=0}^n$ to compute the more reliable h_t as:

$$h_t = E_\theta(u_t^n),\ t \in \mathcal{T}, \tag{1}$$

where θ is the trainable parameters.

Transition module: $f_\phi(h_t)$. After the encoding module outputs the learnable high-resolution state h_t, we then model their dynamics using a neural network. Specifically, $f_\phi : \mathbb{R}^{d_h} \to \mathbb{R}^{d_h}$ to predict the subsequent high-resolution state $h_{t+\tau}$, where d_h defines the dimensionality of the high-resolution state:

$$h_{t+\tau} = f_\phi(h_t),\ t \in \mathcal{T}, \tag{2}$$

where ϕ denotes the trainable parameters.

Down-sampling: $D(h_{t+\tau})$. Please note that data in $\mathcal{D}$ is partial observed, according to Fig. 1 (left), we define a down-sampling operation as $D : \mathbb{R}^{d_h} \to \mathcal{U}$ after the transition module, with the known coordinates. Specifically, for a learnable high-resolution state $h_{t+\tau}$, it is represented as a $n \times n$ matrix where $h_{t+\tau}(x, y)$ denotes the value at coordinates (x, y).As we mentioned in **problem settings**, $\mathcal{X} \subset \Omega$ is the set of the coordinated in which we sample the partial observation. Applying the operation, the predicted partially observed state $\hat{u}_{t+\tau}$ is:

$$\hat{u}_{t+\tau} = \{h_{t+\tau}(x, y) \mid (x, y) \in \mathcal{X}\}. \tag{3}$$

By doing this, we down-sample the subsequent partially observed state $\hat{u}_{t+\tau}$ from $h_{t+\tau}$ in high-resolution space.

Inference: Combined altogether, our components define the inference, shown in Fig. 1 (right), as:

$$\hat{u}_{t+\tau} = D(f_\phi(E_\theta(u_t^n))),\ t \in \mathcal{T}. \tag{4}$$

The usage of encoding module is different during training and inference periods. In training, given the "input-output" pair $(u_t^n, u_{t+\tau}^n)$, it will encode both u_t^n and $u_{t+\tau}^n$ from two ends into h_t and $h_{t+\tau}$ to calculate PDE loss. In inference, only input u_t^n is available, and h_t is computed to downstream transition module. We eliminate the uncertainty caused by missing information in partial observation by containing previous temporal observations into input, which can also be handled by Bayesian neural networks (Louizos and Welling 2017) and VAE (Kingma and Welling 2013), but it is not the focus of this paper.

The "encoding-transition-sampling" framework can leverage the PDE loss to improve the generalization of model only based on partial observation data using following our designed training strategy.

Algorithm 1: RPLPO

Input: Dataset $\mathcal{D}$, $\mathcal{B}$, Parameters θ, ϕ, Gaps between each fine-tuning q, Steps of fine-tuning m_1, m_2.
Initialize θ and ϕ of two modules E_θ, f_ϕ.
while True **do**
  for $i = 1$ to q **do**
    Update θ, ϕ using $\mathcal{L}_D$, $\mathcal{L}_P^\theta$, $\mathcal{L}_P^\phi$; for each $(u_t, u_{t+\tau})$ in $\mathcal{D}$.
  end for
  for $i = 1$ to m_1 **do**
    Update ϕ using $\mathcal{L}_P^\phi$, for each u_t in $\mathcal{B}$.
  end for
  for $i = 1$ to m_2 **do**
    Update θ using $\mathcal{L}_D$, for each $(u_t, u_{t+\tau})$ in $\mathcal{D}$.
  end for
end while

Model Learning

Based on the previous components, there are three problems in training them jointly with data loss and PDE loss. The first is how to train the encoding module when we do not have the data of high-resolution states required by supervised learning. The second is how to use PDE loss to improve generalization. The third is how to use unlabeled data. To solve the above problems, we propose a learning strategy with two periods using the PDE loss function, including a base-training period given the labeled train sequences $\mathcal{D}$ and a two-stage fine-tuning period given the unlabeled data $\mathcal{B}$. We summarize the implementation in Algorithm 1.

Loss functions: We design the loss function with data loss and PDE loss as follows:

$$\begin{aligned}
&\mathcal{L}_{PI}(\theta, \phi, \mathcal{D}) = \mathcal{L}_D(\theta, \phi, \mathcal{D}) + \gamma\mathcal{L}_P^\theta(\theta, \mathcal{D}) + \gamma\mathcal{L}_P^\phi(\phi, \mathcal{D}),\\
&\text{where } \mathcal{L}_D(\theta, \phi, \mathcal{D}) = \frac{\|\hat{u}_{t+\tau} - u_{t+\tau}\|_2}{\|u_{t+\tau}\|_2},\\
&\mathcal{L}_P^\theta(\theta, \mathcal{D}) = F(h_t^\theta, h_{t+\tau}^\theta)^2,\ \mathcal{L}_P^\phi(\phi, \mathcal{D}) = F(h_t^\theta, h_{t+\tau}^\phi)^2,
\end{aligned} \tag{5}$$

where $\mathcal{D}$ is the labeled dataset and $(u_t, u_{t+\tau}) \in \mathcal{D}$; $\mathcal{L}_D$ is the relative l_2 data loss calculated on partial observation. $\mathcal{L}_P^\theta$ and $\mathcal{L}_P^\phi$ denote the PDE losses of encoding module and transition module; γ is the weight of PDE loss; $\hat{u}_{t+\tau}$ is defined in Eqn. 3, h_t^θ and $h_{t+\tau}^\theta$ are outputs of Eqn. 1, $h_{t+\tau}^\phi$ is an output of Eqn. 2. By expressing the finite difference formulae of PDEs following Huang et al. (2023); Ren et al. (2023b) as $F(\psi_t, \psi_{t+\tau}) = 0$, where ψ_t and $\psi_{t+\tau}$ are t and $t + \tau$ solutions of PDEs, PDE losses $\mathcal{L}_P^\theta$ and $\mathcal{L}_P^\phi$ can be represented as

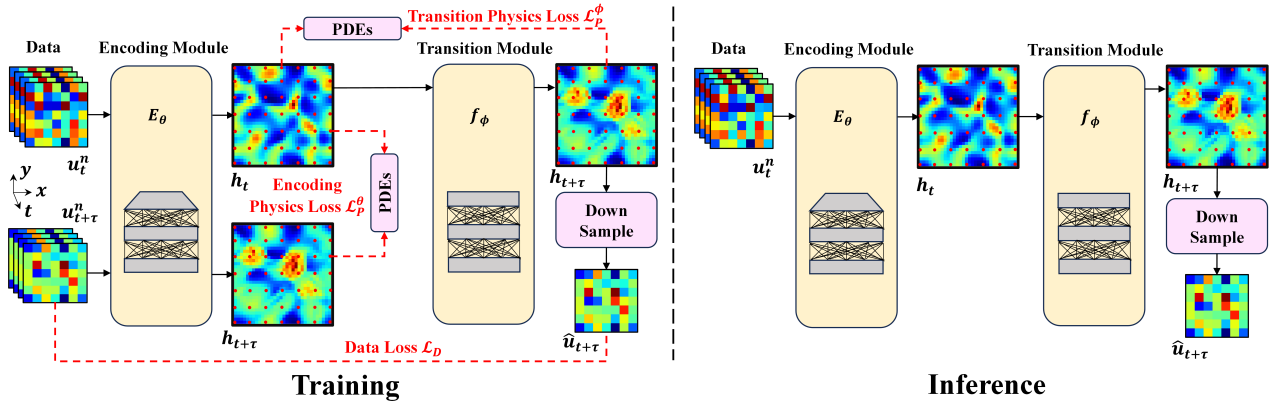

Figure 1: **RPLPO.** Training (left): In the base-training period, the encoding module is trained by $\mathcal{L}_D$ and $\mathcal{L}_P^\theta$, and the transition module is trained by $\mathcal{L}_D$ and $\mathcal{L}_P^\phi$. These losses are calculated on labeled dataset $\mathcal{D}$. Then, in the two-stage fine-tuning period, the transition module is tuned by $\mathcal{L}_P^\phi$, calculated on unlabeled dataset $\mathcal{B}$, and the encoding module is tuned by $\mathcal{L}_D$, calculated on $\mathcal{D}$, in order. Inference (right): given the partial observation to predict future partially observed states. For the two-stage fine-tuning period, please check **Technical Appendix Figure 4**.

Eqn. 5. We apply the widely-used standard 4th-order Runge-Kutta (RK4) and 4th-order central difference scheme in PDE loss calculation.

Base-training period: The base-training period can be formalized as a data loss and a PDE loss joint optimization problem that we solve in parallel:

$$\theta^*, \phi^* = \text{argmin}_{\theta,\phi}\ \mathcal{L}_{PI}(\theta, \phi, \mathcal{D}). \tag{6}$$

Here, we apply distinct training on each module in physics-informed manner, as shown in Fig. 1 (left). **(1)** For training the encoding module to offset the complete lack of fine-grained data, we propose to encode input u_t^n and label $u_{t+\tau}^n$ to learnable high-resolution states h_t and $h_{t+\tau}$ using the same encoding module, which can be used to calculate $\mathcal{L}_P^\theta$ in training. The derivative of $\mathcal{L}_P^\theta$ with respect to the parameter θ in the encoding module is calculated. **(2)** To train the transition module and improve generalization for predicting, $\mathcal{L}_P^\phi$ is computed between the input h_t and the predicted subsequent $h_{t+\tau}$, and the derivative of $\mathcal{L}_P^\phi$ with respect to the parameter ϕ in the transition module is calculated. Along with the above physics-informed manner, the $\mathcal{L}_D$ is also used to train two modules end-to-end.

Two-stage fine-tuning period: We leverage unlabeled data to tune both modules after the base-training period to improve model's generalization further, as shown in Fig. 1 (left). We first optimize ϕ and then optimize θ. The period can be formalized as an optimization problem:

$$\phi^* = \text{argmin}_\phi\ \mathcal{L}_P^\phi(\phi, \mathcal{B}),\ \theta^* = \text{argmin}_\theta\ \mathcal{L}_P^\theta(\theta, \mathcal{D}), \tag{7}$$

where $\mathcal{B}$ is the unlabeled dataset and the definition of $\mathcal{L}_P^\phi$ is same as Eqn. 5, except for $u_t \in \mathcal{B}$. Please note two stages of fine-tuning have an order that is important for optimization. Intuitively, we first fine-tune the transition module with only $\mathcal{L}_P^\phi$ on unlabeled data $\mathcal{B}$ to make h_t and predicted $h_{t+\tau}$ more in line with PDE loss. However, as the encoding module and transition module are first trained via $\mathcal{L}_D$ together in base-training period, the fine-tuned transition module mismatches encoding module now, leading to deteriorating performance in general. Thus, we then fine-tune encoding module with $\mathcal{L}_D$ on $\mathcal{D}$. Because the encoding module has to be trained using both input and label, while the transition module can be trained without labels, we can propagate information of unlabeled data and PDE from transition module to encoding module by using this order.

Experiments

In this section, we validate our framework's generalization capability through experiments, comparing it with baselines for both single and multi-step predictions which is more challenge due to error accumulation. Then, we demonstrate the effectiveness of our framework in terms of data numbers, sparsity level of observation, irregular partial observation, inaccurate PDE, and noisy data. We also assess the framework's adaptability and robustness by varying encoding modules, hyperparameters and necessity of fine-tuning. Finally, we studies the computational cost of our framework under the premise of performance by contrasting cost and GPU memory usage. Due to space limitation, we leave the details of architecture and implementation in **Technical Appendix Model Architecture and Implementation Details**.

Experiments Setup

Benchmarks: We consider five PDEs that can represent common physical systems in real world. Details are in **Technical Appendix Benchmarks**. **(1) Burgers equation** (Burgers) has a one-dimensional output scalar velocity field u. **(2) Wave equation** (Wave) has the two-dimensional output velocity field u and potential field ϕ. **(3) Navier Stokes equation** (NSE) corresponds to incompressible viscous fluid dynamics with three-dimensional output vector velocity (u^x, u^y) and pressure field p. **(4) Linear shallow water equation** (LSWE) corresponds to the inviscid linearized shallow water equation with three-dimensional output vector velocity (u^x, u^y) and height h. **(5) Nonlinear**

Experiments		PIDL	LNPDE	GNOT	PeRCNN	FNO	FNO*	PINO*	**RPLPO**
Burgers	$\mathcal{L}_D$	1.43E-1	1.05E-1	1.93E-2	3.03E-1	1.68E-2	1.69E-2	6.35E-2	**1.37E-2**
	ϵ	9.80E-6	-	-	0.98	-	6.84E-5	9.81E-6	**1.85E-6**
Wave	$\mathcal{L}_D$	1.28	9.93E-1	1.33E-1	5.44E-1	1.11E-1	3.58E-2	1.01	**2.64E-2**
	ϵ	1.19	-	-	1.18	-	2.17	2.09	**1.06**
NSE	$\mathcal{L}_D$	6.45E-1	6.35E-1	1.18E-1	5.07E-1	1.06E-1	1.68E-2	4.70E-1	**1.34E-2**
	ϵ	1.71E-2	-	-	1.07	-	1.85E-2	1.07E-2	**2.76E-8**
LSWE	$\mathcal{L}_D$	7.60	9.26E-2	1.19E-1	4.92E-1	7.92E-2	4.75E-2	7.60	**2.44E-2**
	ϵ	**1.38E-3**	-	-	0.97	-	6.82E-3	1.63E-3	1.47E-3
NSWE	$\mathcal{L}_D$	2.34E-1	6.70E-2	1.63E-1	4.06E-1	1.01E-1	6.41E-2	2.32E-1	**3.50E-2**
	ϵ	3.16E-1	-	-	1.30	-	1.74	3.18E-1	**2.03E-1**

Table 1: **Relative loss $\mathcal{L}_D$ (↓) and ϵ (↓) on five benchmarks.** Our framework achieves better prediction results than all other baseline methods. The results are the mean of three times running with different seeds. Best in **bold** and second best underline.

$\lvert\mathcal{D}\rvert$	FNO*	**RPLPO**	RPLPO w/o FT
50	1.28E-1	**8.81E-2**	9.53E-2
100	1.07E-1	**6.88E-2**	7.34E-2
300	6.41E-2	**3.50E-2**	3.69E-2
350	5.59E-2	**3.21E-2**	3.32E-2

Table 2: Relative loss $\mathcal{L}_D$ (↓) of the data numbers.

shallow water equation (NSWE) is more challenge than LSWE, which retains nonlinear features including uneven bottom. In all benchmarks, ICs are randomly sampled from i.i.d. Gaussian Random Fields and models learn to generalize to various trajectories with different ICs, as detailed in **Technical Appendix Data Generation**.

Baselines: We reimplement several methods as baselines (details in **Technical Appendix Baseline Methods**): **(1) PIDL**, a physics-informed deep learning approach using finite difference-based PDE loss, previously applied in Liu and Wang (2021). **(2) LNPDE** (Iakovlev, Heinonen, and Lähdesmäki 2023), a state-of-the-art model for learning physical dynamics, independent of the space-time continuous grid. **(3) GNOT** (Hao et al. 2023), an effective transformer for learning operators. **(4) PeRCNN** (Rao et al. 2023), a framework encoding physics to learn spatiotemporal dynamics. **(5) FNO** (Li et al. 2020b), a neural operator with FFT-based spectral convolutions. **(6) FNO***, enhancing FNO by attaching the same encoding module as in RPLPO, with $\gamma = 0$ in Eqn. 5. **(7) PINO** (Li et al. 2021b), a hybrid approach merging data and PDE loss like FNO*, calculating PDE loss between u_t and $u_{t+\tau}$. Attaching the same encoding module in FNO* and PINO* levels the playing field to evaluate our method versus modified baselines.

	FNO*	**RPLPO**
w/o high-resolution states	6.41E-2	**3.50E-2**
with 1/3 high-resolution states	5.13E-2	**3.41E-2**
with 1/2 high-resolution states	5.01E-2	**3.35E-2**
with all high-resolution states	4.70E-2	**3.29E-2**

Table 3: Relative loss $\mathcal{L}_D$ (↓) of the number of corresponding high-resolution states if available.

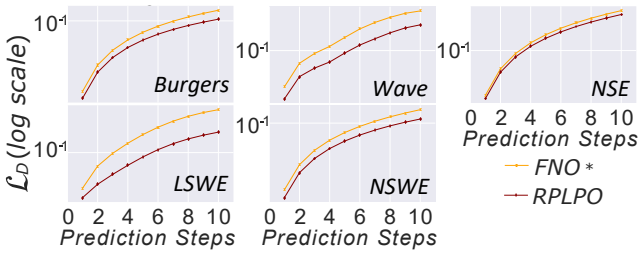

Figure 2: **The results of multi-steps prediction from 1st to 10th step.** RPLPO achieves a significant improvement against FNO* on five benchmarks at all prediction steps.

Comparison With Baselines

We compare our proposed RPLPO with seven baselines; the results are shown in Table. 1 for single-step prediction. We visualize some prediction results in **Technical Appendix Result Visualization**. Specifically, RPLPO shows a significant improvement against all baselines on five benchmarks. RPLPO has at least 25% (against FNO* in NSE) to more than 99% (against PINO* in LSWE) improvement on $\mathcal{L}_D$ compared with all baselines. Moreover, RPLPO simultaneously has over 36% (against PINO* in NSWE) improvement on ϵ than most of the baselines, which means RPLPO can reconstruct the more accurate high-resolution state. We consider it a reason why RPLPO can learn the physical system models with better generalization. Note that in LSWE, reconstruction error ϵ of PIDL is slightly better than ours, but its data loss $\mathcal{L}_D$ is much larger than ours because PIDL focuses on training the model with PDE loss only, ignoring the data constraint for predicting. In general, RPLPO achieves better performance than all baselines, which demonstrates its modeling and generalization.

Furthermore, we evaluate our framework on the multi-step prediction and compared it with baselines. The results are shown in Fig. 2 against FNO* (the best method among baselines), and more detailed results of others are left in **Technical Appendix Multi-step Prediction**. Our proposed RPLPO achieves a significant improvement against FNO* on all benchmarks among prediction steps. In general, the gap between $\mathcal{L}_D$ of RPLPO and FNO* is increasing along

$x_u \times y_u$	FNO*	**RPLPO**
3×3	7.86E-2	**4.43E-2**
5×5	7.08E-2	**3.30E-2**
7×7	6.41E-2	**3.50E-2**

Table 4: Relative loss $\mathcal{L}_D$ (↓) of different sparsity levels of partial observation.

MP-PDE	MeshGraphNets	FNO*	**RPLPO**
3.44E-1	1.85E-1	7.49E-2	**5.03E-2**

Table 5: Relative loss $\mathcal{L}_D$ (↓) of irregular observation.

with the prediction steps. The reason is that the growth of cumulative error may slow down due to the model being constrained to meet the PDE loss at each step, thus maintaining higher generalization in multi-step prediction.

Ablation Studies

We conduct comprehensive experiments to verify the effectiveness, robustness, and computational cost of RPLPO. The additional ablation studies, including zero-shot super-resolution, types of data loss, finite-difference schemes of PDE loss, and physics metrics, are available in the **Technical Appendix**.

Effectiveness In this subsection, we demonstrate how RPLPO effectively operates across data numbers, sparsity level of observation, irregular partial observation, inaccurate PDE, and noisy data.

Ablation study on data numbers: We evaluate the impact of data numbers for the partial observation $\mathcal{D}$ and high-resolution states, if available, by experiments on NSWE. **(1)** Using different numbers of trajectory $|\mathcal{D}|$, and **(2)** assuming almost 1/3, 1/2, and all partial observations have corresponding high-resolution states used as labels for the encoding module, allowing more accurate state reconstruction to explore if they can improve the performance.

(i) Table 2 shows that $\mathcal{L}_D$ decreases as $|\mathcal{D}|$ increases; larger data volumes enhance model generalization. With less data, our framework still outperforms FNO*, although it's more effective with more data. **(ii)** Table 3 reveals that both FNO* and RPLPO benefit from increased high-resolution states, with RPLPO showing greater improvements. Please note that the relative improvement of RPLPO over FNO* decreases with the increase of high-resolution states, indicating RPLPO's function in compensating for the lack of high-resolution states.

Sparsity level of observation and irregular observation: We consider the impact of sparsity level and irregularity in partial observations. **(1)** Using different sparsity levels of observation data $x_u \times y_u$ in training. **(2)** Using irregular partial observation, sampled randomly from high-resolution states as shown in **Technical Appendix Fig. 5** to verify the performance on irregular observation.

(i) Table 4 shows improved performance with relatively more observation ($5 \times 5, 7 \times 7$) during training, are more significant. Across various $x_u \times y_u$, RPLPO consistently outperforms FNO*. **(ii)** For experiments on irregular observation, as shown in Table 5, we compared RPLPO with two popular methods of irregular data MP-PDE (Brandstetter, Worrall, and Welling 2022) and MeshGraphNets (Pfaff et al. 2020). RPLPO can improve performance over the baselines, consistent with the conclusion on the regular observations.

	0	10%	20%	30%
Accurate PDE	3.50E-2	5.05E-2	6.11E-2	6.80E-2
GRFs with std 1	3.52E-2	5.06E-2	6.16E-2	6.76E-2
GRFs with std 5	4.00E-2	5.44E-2	6.30E-2	6.91E-2
GRFs with std 10	5.12E-2	5.92E-2	6.79E-2	8.50E-2
FNO*	6.41E-2	8.47E-2	8.98E-2	9.66E-2

Table 6: Relative loss $\mathcal{L}_D$ (↓) of noisy data and inaccurate PDE. The column means the noise percentages (10%, 20%, 30%) on observed data. 0 means data do not have noise. The row means baseline FNO* and different scales GRFs added on the PDE used in the RPLPO.

Architecture	FNO*	**RPLPO**
U-Net	6.41E-2	**3.50E-2**
Transformer	6.79E-2	**4.70E-2**

Table 7: Relative loss $\mathcal{L}_D$ (↓) using U-Net and Transformer.

Inaccurate PDE and noisy data: To simulate the **challenges in real-world practices**, including noisy data and inaccurate PDE, we consider the above two challenges to evaluate the performance of RPLPO. We add the unknown terms as the Gaussian random fields (GRFs) in PDE to represent the inaccuracy and use the data with different levels (10%, 20%, and 30%) Gaussian noise following the previous work (Rao et al. 2023). We use the GRFs with mean 0 and std 1, 5, and 10. From the results in Table 6, we can see that RPLPO improves performance against the baseline in all settings. It illustrates that our method is effective even when the PDE is not accurate and the data has noise.

Error	FNO*	NN	bilinear	bicubic	**RPLPO**
ϵ	1.74	8.40E-1	5.98E-1	5.83E-1	**2.03E-1**

Table 8: Relative reconstruction error ϵ (↓) of the proposed encoding module and interpolations.

Computational Cost and Analysis

Adaptability and Robustness We change the network architecture of encoding module, hyperparameters, and RPLPO without fine-tuning to verify the framework always has improvement under different implementations.

Network architecture of encoding module: To demonstrate RPLPO's independence from specific network architectures, we substitute the encoding module's U-Net with a Transformer in the NSWE setting, using ViT's official Transformer implementation (Dosovitskiy et al. 2020). Ta-

Method	Inference (s)			Training (s)			Inference (MiB)			Training (MiB)		
	32	48	64	32	48	64	32	48	64	32	48	64
FNO*	**0.0751**	0.1445	0.2387	**0.4555**	**1.5786**	**2.8061**	**3479**	**3955**	**5445**	**4311**	**5731**	**8457**
RPLPO	0.0764	**0.1398**	**0.2374**	0.8007	2.5895	3.8649	**3479**	**3955**	**5445**	4383	5851	8615

Table 9: The increasing of computational cost (s ↓) and GPU memory utilize (MiB ↓) along the resolutions of high-resolution states increase in inference and training. Headers are three resolutions, like 64×64.

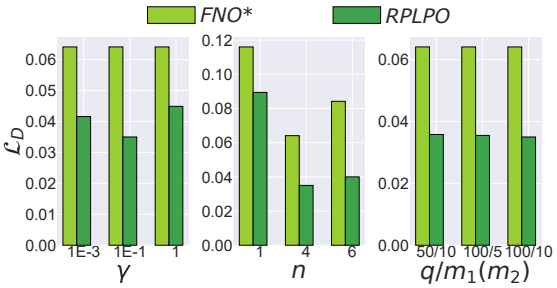

Figure 3: Relative loss $\mathcal{L}_D(\downarrow)$ of hyperparameters. In the right one, we omit "m_2" on x-coordinate means $m_2 = m_1$.

ble 7 shows that RPLPO can still significantly outperforms the data-driven approach FNO* by using PDE loss, particularly when fine-tuning both the U-Net and Transformer encoding modules. Moreover, we replace the PDE loss-trained encoding module with interpolation methods, including nearest neighbors (NN), bilinear, and bicubic interpolations. The result, shown in Table 8, verifies the necessity and effectiveness of the proposed encoding module.

Ablation study on hyperparameters: We examine three critical hyperparameters in RPLPO for robustness: **(1)** PDE loss weight γ from Eqn. 5, **(2)** lengths of recent temporal observations n as input, and **(3)** the number of epochs m_1, m_2 and gaps q between each fine-tuning period. **(i)** Figure 3 (left) indicates clearer model improvement at γ = 1E-1, chosen for this work. RPLPO shows consistent enhancement regardless of γ, demonstrating robustness. **(ii)** Longer n values ($n = 4, 6$) lead to better RPLPO performance, as seen in Figure 3 (middle), by utilizing more temporal observations. **(iii)** Figure 3 (right) shows RPLPO's improvements across all parameters, confirming RPLPO can adapt to varying training setups. We also study the impact of two PDE losses with different γ and the higher upscale factor of the encoding module. Their results are shown in **Technical Appendix Table 18 and Table 24**.

Necessity of fine-tuning: The two-stage fine-tuning period is an important part of RPLPO as the data scarcity tend to cause the generalization issues, especially when data acquisition is costly and only partial observation is available. We conduct the ablation study on the relationship between the data and the effect of fine-tuning. In Table 2, the RPLPO w/o FT means RPLPO is not trained by the two-stage fine-tuning period. RPLPO has greater improvement than RPLPO w/o FT when data is limited, as leveraging the unlabeled data provides additional information.

We design experiments to study the computational cost and GPU usage impacted by introducing PDE loss, and cost on the transition module on a single Nvidia V100 16GB GPU under the premise of the above effectiveness, adaptability, and robustness to demonstrate the efficiency of RPLPO.

Method	32 (s)	48 (s)	64 (s)
f_ϕ **of RPLPO**	**0.0030**	**0.0032**	**0.0032**
Solver	0.0159	0.0168	0.0175

Table 10: Computational cost (s ↓) of our proposed transition module and numerical solver. Headers are three resolutions of high-resolution states, like 64×64.

Impact of PDE loss on cost and GPU usage: We assess the rise in computational cost along the increasing of resolutions of the learnable high-resolution state. Table 9 shows that regardless of resolution, FNO* and RPLPO incur similar costs and GPU memory usage due to identical model structures. However, RPLPO's training costs are higher than FNO*'s due to differing loss functions, involving calculations of PDE loss and its derivatives across two modules, but these costs remain within a reasonable range.

Cost on transition module: We study the computational cost of our transition module against the numerical solver (RK4) to verify the efficiency and necessity of our proposed component. As shown in Table 10, the cost taken for a step forward using solver is more than five times that of our transition module. Moreover, as the resolution increases, the computational cost of our transition module does not significantly change, as only the input and output layers are affected, with small changes in the hidden layers. These results indicate that, in terms of computational cost, whether in training or inference, our transition module offers considerable advantages over the solver. Detail is available in **Technical Appendix Cost on Transition Module**.

Conclusion

In this paper, we propose RPLPO, a novel framework to re-enable PDE loss under partial observation to improve the model's generalization for predicting future partially observed states. Within RPLPO, we use the encoding module and transition module in order and develop a training strategy with two periods to address challenges associated with partial observation and data scarcity. Using five physical systems as examples, we demonstrate that RPLPO can successfully re-enable PDE loss and thus improve the model's generalization capacity. **Technical Appendix is in the extended version.**

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant NO. 12302365).

References

An, S. S.; Kim, T.; and James, D. L. 2008. Optimizing cubature for efficient integration of subspace deformations. *ACM transactions on graphics (TOG)*, 27(5): 1–10.

Boussif, O.; Bengio, Y.; Benabbou, L.; and Assouline, D. 2022. MAgnet: Mesh agnostic neural PDE solver. *Advances in Neural Information Processing Systems*, 35: 31972–31985.

Brandstetter, J.; Worrall, D.; and Welling, M. 2022. Message passing neural PDE solvers. *arXiv preprint arXiv:2202.03376*.

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

Chen, P. Y.; Xiang, J.; Cho, D. H.; Yue Chang, G. A., Pershing; Maia, H. T.; Chiaramonte, M. M.; Carlberg, K. T.; and Grinspun, E. 2023. CROM: Continuous Reduced-Order Modeling of PDEs Using Implicit Neural Representations. In *International Conference on Learning Representations*.

Dong, C.; Loy, C. C.; He, K.; and Tang, X. 2014. Learning a deep convolutional network for image super-resolution. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, 184–199. Springer.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Esmaeilzadeh, S.; Azizzadenesheli, K.; Kashinath, K.; Mustafa, M.; Tchelepi, H. A.; Marcus, P.; Prabhat, M.; Anandkumar, A.; et al. 2020. Meshfreeflownet: A physics-constrained deep continuous space-time super-resolution framework. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, 1–15. IEEE.

Fathi, M. F.; Perez-Raya, I.; Baghaie, A.; Berg, P.; Janiga, G.; Arzani, A.; and D'Souza, R. M. 2020. Super-resolution and denoising of 4D-flow MRI using physics-informed deep neural nets. *Computer Methods and Programs in Biomedicine*, 197: 105729.

Gao, H.; Sun, L.; and Wang, J.-X. 2021a. PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain. *Journal of Computational Physics*, 428: 110079.

Gao, H.; Sun, L.; and Wang, J.-X. 2021b. Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. *Physics of Fluids*, 33(7).

Gao, S.; Liu, X.; Zeng, B.; Xu, S.; Li, Y.; Luo, X.; Liu, J.; Zhen, X.; and Zhang, B. 2023. Implicit diffusion models for continuous super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10021–10030.

Goswami, S.; Bora, A.; Yu, Y.; and Karniadakis, G. E. 2022a. Physics-informed deep neural operators networks. *arXiv preprint arXiv:2207.05748*.

Goswami, S.; Yin, M.; Yu, Y.; and Karniadakis, G. E. 2022b. A physics-informed variational DeepONet for predicting crack path in quasi-brittle materials. *Computer Methods in Applied Mechanics and Engineering*, 391: 114587.

Gupta, G.; Xiao, X.; and Bogdan, P. 2021. Multiwavelet-based operator learning for differential equations. *Advances in neural information processing systems*, 34: 24048–24062.

Hansen, D.; Maddix, D. C.; Alizadeh, S.; Gupta, G.; and Mahoney, M. W. 2023. Learning physical models that can respect conservation laws. In *International Conference on Machine Learning*, 12469–12510. PMLR.

Hao, Z.; Wang, Z.; Su, H.; Ying, C.; Dong, Y.; Liu, S.; Cheng, Z.; Song, J.; and Zhu, J. 2023. Gnot: A general neural operator transformer for operator learning. In *International Conference on Machine Learning*, 12556–12569. PMLR.

Huang, X.; Shi, W.; Meng, Q.; Wang, Y.; Gao, X.; Zhang, J.; and Liu, T.-Y. 2023. NeuralStagger: accelerating physics-constrained neural PDE solver with spatial-temporal decomposition. *arXiv preprint arXiv:2302.10255*.

Iakovlev, V.; Heinonen, M.; and Lähdesmäki, H. 2020. Learning continuous-time pdes from sparse data with graph neural networks. *arXiv preprint arXiv:2006.08956*.

Iakovlev, V.; Heinonen, M.; and Lähdesmäki, H. 2023. Learning Space-Time Continuous Neural PDEs from Partially Observed States. *arXiv preprint arXiv:2307.04110*.

Jangid, D. K.; Brodnik, N. R.; Goebel, M. G.; Khan, A.; Majeti, S.; Echlin, M. P.; Daly, S. H.; Pollock, T. M.; and Manjunath, B. 2022. Adaptable physics-based super-resolution for electron backscatter diffraction maps. *npj Computational Materials*, 8(1): 255.

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

Kelshaw, D.; Rigas, G.; and Magri, L. 2022. Physics-informed CNNs for super-resolution of sparse observations on dynamical systems. *arXiv preprint arXiv:2210.17319*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4681–4690.

Li, K.; Li, C.; Yu, B.; Shen, Z.; Zhang, Q.; He, S.; and Chen, J. 2021a. Model and transfer spatial-temporal knowledge for fine-grained radio map reconstruction. *IEEE Transactions*

on Cognitive Communications and Networking, 8(2): 828–841.

Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; and Anandkumar, A. 2020a. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*.

Li, Z.; Kovachki, N. B.; Azizzadenesheli, K.; Bhattacharya, K.; Stuart, A.; Anandkumar, A.; et al. 2020b. Fourier Neural Operator for Parametric Partial Differential Equations. In *International Conference on Learning Representations*.

Li, Z.; Zheng, H.; Kovachki, N.; Jin, D.; Chen, H.; Liu, B.; Azizzadenesheli, K.; and Anandkumar, A. 2021b. Physics-informed neural operator for learning partial differential equations. *arXiv preprint arXiv:2111.03794*.

Lim, B.; Son, S.; Kim, H.; Nah, S.; and Mu Lee, K. 2017. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 136–144.

Liu, X.-Y.; and Wang, J.-X. 2021. Physics-informed Dyna-style model-based deep reinforcement learning for dynamic control. *Proceedings of the Royal Society A*, 477(2255): 20210618.

Liu, Z.-S.; Siu, W.-C.; and Wang, L.-W. 2021. Variational autoencoder for reference based image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 516–525.

Louizos, C.; and Welling, M. 2017. Multiplicative normalizing flows for variational bayesian neural networks. In *International Conference on Machine Learning*, 2218–2227. PMLR.

Lu, L.; Jin, P.; and Karniadakis, G. E. 2019. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*.

Lütjens, B.; Crawford, C. H.; Watson, C. D.; Hill, C.; and Newman, D. 2022. Multiscale neural operator: Learning fast and grid-independent pde solvers. *arXiv preprint arXiv:2207.11417*.

Nazeri, K.; Thasarathan, H.; and Ebrahimi, M. 2019. Edge-informed single image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 0–0.

Pfaff, T.; Fortunato, M.; Sanchez-Gonzalez, A.; and Battaglia, P. W. 2020. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*.

Raissi, M.; Perdikaris, P.; and Karniadakis, G. E. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378: 686–707.

Rao, C.; Ren, P.; Wang, Q.; Buyukozturk, O.; Sun, H.; and Liu, Y. 2023. Encoding physics to learn reaction–diffusion processes. *Nature Machine Intelligence*, 5(7): 765–779.

Ren, P.; Erichson, N. B.; Subramanian, S.; San, O.; Lukic, Z.; and Mahoney, M. W. 2023a. Superbench: A super-resolution benchmark dataset for scientific machine learning. *arXiv preprint arXiv:2306.14070*.

Ren, P.; Rao, C.; Liu, Y.; Ma, Z.; Wang, Q.; Wang, J.-X.; and Sun, H. 2023b. PhySR: Physics-informed deep super-resolution for spatiotemporal data. *Journal of Computational Physics*, 492: 112438.

Shu, D.; Li, Z.; and Farimani, A. B. 2023. A physics-informed diffusion model for high-fidelity flow field reconstruction. *Journal of Computational Physics*, 478: 111972.

Soh, J. W.; Park, G. Y.; Jo, J.; and Cho, N. I. 2019. Natural and realistic single image super-resolution with explicit natural manifold discrimination. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8122–8131.

Wang, C.; Bentivegna, E.; Zhou, W.; Klein, L.; and Elmegreen, B. 2020. Physics-informed neural network super resolution for advection-diffusion models. *arXiv preprint arXiv:2011.02519*.

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

Wang, S.; Wang, H.; and Perdikaris, P. 2021. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Science advances*, 7(40): eabi8605.

Yang, L.; Meng, X.; and Karniadakis, G. E. 2021. B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *Journal of Computational Physics*, 425: 109913.

Yin, Y.; Kirchmeyer, M.; Franceschi, J.-Y.; Rakotomamonjy, A.; and Gallinari, P. 2022. Continuous pde dynamics forecasting with implicit neural representations. *arXiv preprint arXiv:2209.14855*.

Yin, Y.; Kirchmeyer, M.; Franceschi, J.-Y.; Rakotomamonjy, A.; and Gallinari, P. 2023. Continuous PDE Dynamics Forecasting with Implicit Neural Representations. In *The Eleventh International Conference on Learning Representations*.

Zayats, M.; Zimoń, M. J.; Yeo, K.; and Zhuk, S. 2022. Super Resolution for Turbulent Flows in 2D: Stabilized Physics Informed Neural Networks. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, 3377–3382. IEEE.

Zhang, X.; Wang, L.; Helwig, J.; Luo, Y.; Fu, C.; Xie, Y.; Liu, M.; Lin, Y.; Xu, Z.; Yan, K.; et al. 2023. Artificial Intelligence for Science in Quantum, Atomistic, and Continuum Systems. *arXiv preprint arXiv:2307.08423*.

Zhao, H.; Kong, X.; He, J.; Qiao, Y.; and Dong, C. 2020. Efficient image super-resolution using pixel attention. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, 56–72. Springer.

Zhu, X.; Yang, F.; Huang, D.; Yu, C.; Wang, H.; Guo, J.; Lei, Z.; and Li, S. Z. 2020. Beyond 3dmm space: Towards fine-grained 3d face reconstruction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, 343–358. Springer.