Breaking the Discretization Barrier of Continuous Physics Simulation Learning

Fan Xu[1*], **Hao Wu**[2*], **Nan Wang**[3†], **Lilan Peng**[4], **Kun Wang**[5†], **Wei Gong**[1†], **Xibin Zhao**[2†]
[1]University of Science and Technology of China [2]Tsinghua University
[3]Beijing Jiaotong University [4]Southwest Jiaotong University [5]Nanyang Technological University
[*] Equal Contribution, [†] Corresponding author
Main Contact: markxu@mail.ustc.edu.cn

Abstract

The modeling of complicated time-evolving physical dynamics from partial observations is a long-standing challenge. Particularly, observations can be sparsely distributed in a seemingly random or unstructured manner, making it difficult to capture highly nonlinear features in a variety of scientific and engineering problems. However, existing data-driven approaches are often constrained by fixed spatial and temporal discretization. While some researchers attempt to achieve spatio-temporal continuity by designing novel strategies, they either overly rely on traditional numerical methods or fail to truly overcome the limitations imposed by discretization. To address these, we propose `CoPS`, a purely data-driven methods, to effectively model continuous physics simulation from partial observations. Specifically, we employ multiplicative filter network to fuse and encode spatial information with the corresponding observations. Then we customize geometric grids and use message-passing mechanism to map features from original spatial domain to the customized grids. Subsequently, `CoPS` models continuous-time dynamics by designing multi-scale graph ODEs, while introducing a Markov-based neural auto-correction module to assist and constrain the continuous extrapolations. Comprehensive experiments demonstrate that `CoPS` advances the state-of-the-art methods in space-time continuous modeling across various scenarios. The source code is available at https://github.com/Sunxkissed/CoPS.

1 Introduction

Achieving accurate global modeling and forecasting of a complex time-evolving physical system from a limited number of observations has been a long-standing challenge [5, 52, 11]. This has widespread applications in chaotic physical systems such as geophysics [12, 38], atmospheric science [9, 40], and fluid dynamics [56, 29, 55, 4], *et al*. For instance, in meteorology and oceanography, sensors are often deployed in areas with scarce or non-existent network connectivity. This renders the analysis and modeling of the system a formidable obstacle. From empirical knowledge, dynamical systems can be fully described by complicated and not yet fully elucidated partial differential equations (PDEs) [16]. Traditional numerical methods such as finite element [13] and finite volume [2] methods excessively rely on prior knowledge of essential PDEs and suffer from high time complexity, making it difficult to accommodate the increasing demands for grid granularity. Recently, data-driven approaches partially overcome these by integrating advanced neural networks to directly learn dynamic latent features with high nonlinearity from existing observations or simulations.

Although data-driven methods [18, 31, 45] have the capacity to learn complex relationships from observations, they still have serious limitations in changeable actual scenes. One significant challenge in modeling physical fields from partial observations is that the field may not adhere to a Cartesian

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

grid [34, 1], and convolutional neural networks are inherently not adapted to such structures. When employing graph neural networks [22, 51], the graph topology is often determined by the locations of existing observation points, making the model difficult to generalize to unseen spatial locations. Additionally, some methods use padding or interpolation [6] to enforce spatiotemporal continuity. This may distort the inherent inductive bias of the observed data and significantly increases the computational burden in sparse data scenarios.

Recently, to better explore the space-time continuous formulation from partial observations [19, 53], a few notable works stand out. Example of continuous time and space dynamic system evolution is illustrated in Figure 1. For space-continuous learning, the multiplicative filter network (MFN) [10], combined with implicit neural representations (INRs) [17], has been proposed to incorporate coordinate information through linear Fourier or wavelet functions in an efficient manner. However, this approach requires iterating for hundreds of steps to obtain the corresponding implicit neural representation for new samples, severely lacking real-time applicability. Further, for continuous-time learning, neural ordinary differential equations (Neural ODEs) [7] address the limitation of fixed time extrapolation windows by learning continuous ODEs from discrete data using explicit solvers such as the Euler or Runge-Kutta methods [3, 33]. However, if the system's nonlinearity is high, Neural ODE may struggle to capture the essential features of the complex dynamics. Additionally, Neural ODE is solved through numerical integration, which leads to errors accumulating as time progresses. As a result, conventional Neural ODEs may lead to poor stability in long-term predictions and potentially worse fitting.

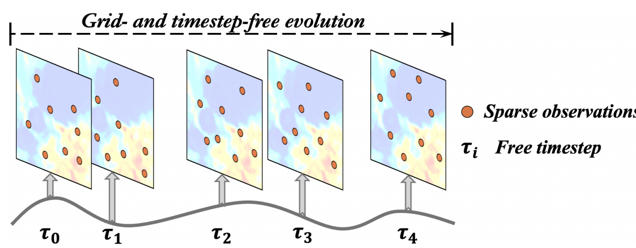

Figure 1: Example of continuous time and space dynamic system evolution. (*Non-discretized*)

To address these, we propose `CoPS`, a purely data-driven methods, to effectively achieve space-time continuous prediction of dynamical systems based on partial observations. Specifically, we employ multiplicative filter networks to fuse and encode spatial information with the corresponding observations. Then we customize geometric grids and use message passing mechanism to map features from original spatial domain to the customized grids. Subsequently, `CoPS` models continuous dynamics by designing multi-scale Graph ODEs, while introducing a local refinement feature extractor to assist and constrain the parametric evolutions. Finally, the predicted latent features are mapped back to original spatial domain through a GNN decoder.

In summary, we make the following key contributions: ① ***Encoding Mechanism.*** We propose a novel encoding approach to integrate partial observations with spatial coordinate information, effectively encoding these features into a customized geometric grid. ② ***Novel Methodology.*** We introduce a multi-scale Graph ODE module to model continuous-time dynamics, complemented by a neural auto-regressive correction module to assist and constrain the parametric evolution, ensuring robust and accurate predictions. ③ ***Superior Performance.*** Our method demonstrates state-of-the-art performance on complex synthetic and real-world datasets, showcasing the possibility for accurate and efficient modeling of intricate dynamical processes and precise long-term predictions.

2 Related Work

2.1 Deep Learning for Physical Simulations

Recent advancements in deep neural networks have established them as effective tools for addressing the complexities of dynamics modeling [15, 52, 50], demonstrating their ability to efficiently capture the intricacies of high-dimensional systems. Physics-Informed Neural Networks (PINNs) [21, 44, 25], which optimize weights by minimizing the residual loss derived from the PDE, have received considerable attention due to their flexibility in tackling a wide range of data-driven solutions [27, 46]. Recently, neural operators [24, 29, 49, 35] map between infinite–dimensional function spaces by replacing standard convolution with continuous alternatives. Specifically, they utilize kernels in fourier space [29, 28, 42] or graph structure [30, 54, 57, 14] to learn the correspondence from the initial condition to the solution at a fixed horizon. However, most existing methods are limited by static observation grids, restricting their ability to evaluate points outside training grids and

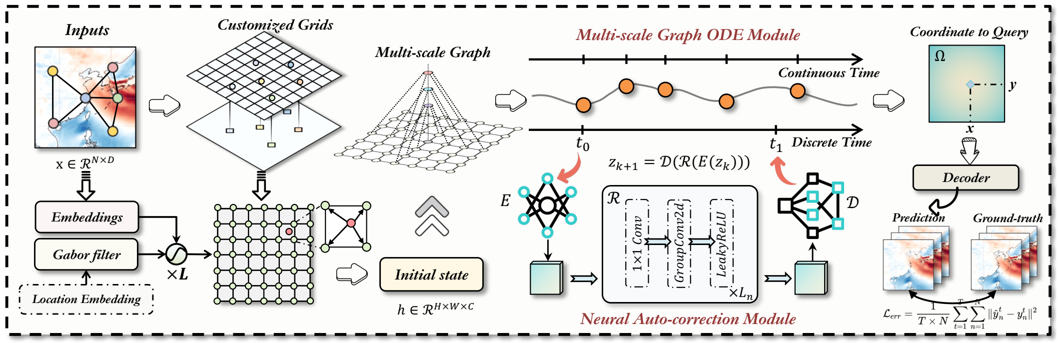

Figure 2: The overview of `CoPS`. *Stage 1:* Employ multiplicative filter network to encode the initial representation, and map it to the customized grids with message passing scheme. *Stage 2:* Model the latent dynamics with multi-scale Graph ODE module and auto-correction module in a continuous-time way. *Stage 3:* Extrapolate results for arbitrary future time step and coordinates.

confining queries to fixed time intervals. In this work, we aim to overcome discretization and achieve spatiotemporal modeling in a continuous way.

2.2 Advanced Space-Continuous Modeling

Interpolation has been widely adopted in numerous applications [6, 20, 47], which also holds wide prospects in spatiotemporal modeling. For instance, researchers employed interpolation as a post-processing technique to generate continuous predictions. Recently, MAgNet [5] proposes to predict the dynamical solution after interpolating the observation graph in latent space to new query points. Another approach for space-continuous or grid-independence modeling is a kind of coordinate-based neural networks, called Implicit Neural Representations (INRs) [17, 23]. Typically, INRs take the spatial coordinates as inputs along with other conditions, which can be utilized to enhance real space-continuous modeling and dynamical evolution predicting. DINO [52] overcomes the limitation of failing to generalize to new grids or resolution via a continuous dynamics model of the underlying flow. MMGN [32] learns relevant basis functions from sparse observations to obtain continuous representations after factorizing spatiotemporal variability into spatial and temporal components.

3 Methodology

Problem Definition. In this work, we focus on modeling spatiotemporal dynamical systems through partial observations. The systems are defined over the temporal domain $\mathcal{T}$ and spatial domain Ω. The observations are recorded at L arbitrary consecutive time points $t_{1:L} := (t_1, \ldots, t_L) \in \mathcal{T}$, and N arbitrary local sensor measurements at locations $x_i \in \Omega, i = (1, \ldots, N)$. To adapt to general real-world scenarios, we assume that all dynamics are learned from the initial observation $u(t_0) = (u(x_1, t_0), \ldots, u(x_N, t_0))^T$. Moreover, the values of time steps and observation locations may vary across different observed trajectories during the inference phase. Thus, for prediction, our goal is to learn a neural function $\mathcal{Q}(\cdot)$, which maps the initial observation at the first time step to future dynamical predictions at arbitrary time steps t_i and locations x_j.

$$\mathcal{Q}(u(t_0, x_0); \mathcal{T}, \Omega) \to \mathcal{Y}(t_i, x_j), \quad t_i \in \mathcal{T},\ x_j \in \Omega. \tag{1}$$

Framework Overview. In this section, we introduce the proposed `CoPS`, which achieves the continuous spatiotemporal modeling from partial observations within complicated dynamical systems. We aim to simultaneously overcome the discretization limitations in both the spatial and temporal domains. To better align with the form of numerical solutions, we begin with partial observations of the initial state and apply constraints based on real observations during the continuous evolution process. An overview of our `CoPS` can be seen in Figure 2.

3.1 Initial Encoding via Multiplicative Filter Network

Initial Encoder. Further, we follow the multiplicative filter network and choose nonlinear Gabor filter [10] $g_\kappa(\cdot)$ to generate a continuous global frequency representation. The formulation is as follows:

$$g_\kappa(x) = \exp\left(-\frac{\gamma^{(\kappa)}}{2}\left\|x - \mu^{(\kappa)}\right\|_2^2\right)\sin\left(W_g^{(\kappa)}x + b_g^{(\kappa)}\right), \tag{2}$$

where $\mu^{(\kappa)} \in \mathbb{R}^{d_h}$ and $\gamma^{(\kappa)} \in \mathbb{R}^{d_h \times d_x}$ denote the respective mean and scale term of the κ-th Gabor filter. Subsequently, the filters are multiplied by the linear transformation of previous layer's embedding and the historical sequence representation u_i. The iteration process is defined as follows:

$$\begin{gathered}\rho_i^1 = g_1(x_i) \;\;\Rightarrow\;\; \rho_i^{(\kappa+1)} = (u_i + W_i^\kappa \rho_i^\kappa + b_i^\kappa) \odot g_\kappa(x), \;\; \text{for } \kappa = 1, \ldots, \mathcal{M}-1, \\ \mathcal{H}_\omega(u_i, x_i) = u_i + W_i^{\mathcal{M}} \rho_i^{\mathcal{M}} + b_i^{\mathcal{M}},\end{gathered} \tag{3}$$

where $\odot$ denotes element-wise multiplication, W_i^κ and b_i^κ are the learnable parameters. Finally, $\mathcal{H}_\omega(u_i, x_i)$ is the obtained vector that contain coordinate information and feature of node i.

Customized Grids and Message Passing Scheme. For our observational data, the spatial domain is non-discrete. Recent works like MeshGraphNet [34] rely on fixed meshes, which present challenges when generalizing to previously unseen coordinate points. We propose a novel customized grid mapping approach. Specifically, for any arbitrary coordinate, it can be mapped onto a uniform grid structure. Let $p_i = (x_i, y_i)$ denote an arbitrary point in the continuous $2D$ space, we will discover the appropriate grid cell according to it. To enrich the representation, we connect each point p_i to the four vertices $\{v_{i1}, v_{i1}, v_{i3}, v_{i4}\}$ of the associated grid cell.

Based on the encoded initial state $h_i^0 = \mathcal{H}_\omega(u_i, x_i)$ and the customized grids, the next step involves a message-passing scheme that propagates the features from the original continuous space to the customized grids. To further preserve spatial continuity, the scheme encodes and incorporates the relative positions of the connected points during message passing. The transformation is as follows:

$$h_i^{k+1} = h_i^k + \sum_{j \in \mathcal{N}(p_i)} W_{ij}^k \left(h_j^k - h_i^k + \phi(x_i, x_j)\right) + b^k, \tag{4}$$

where W_{ij}^k and b^k are learnable parameters, and $\phi(x_i, x_j)$ denotes the relative position embedding of node i and j. In this way, we collaboratively encode both the feature information and rich spatial information onto the customized grids, completing the initial encoding process.

3.2 Latent Dynamics Modeling

Further, we view high-dimensional features on the customized grids as latent states and aim to efficiently model complicated dynamics in a continuous-time way.

Multi-scale Graph ODE. To capture spatiotemporal dynamics over complex domains, we build a hierarchical graph representation inspired by GraphCast [26] on top of the customized grids. Our objective is to progressively encode both coarse global behaviors and refined local structures. Concretely, let $\{\mathcal{G}^{(s)}\}_{s=1}^S$ be a collection of graphs, where $\mathcal{G}^{(1)}$ coincides with the finest-scale grid obtained from the initial encoding process. Specifically, within each graph $\mathcal{G}^{(s)}$, we connect each node to its spatial neighbors by "jump adjacency" on the underlying $2D$ grid, thus $\mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(S)}$ providing graph topologies with longer-range dependencies.

To perform hierarchical message passing, we process each scale independently and then fuse the features using an attention mechanism. For each scale s, the message passing is defined as follows:

$$h_i^{'(l,s)} = \text{AGGREGATE}\left(\left\{h_j^{(l-1,s)} : j \in \mathcal{N}^{(s)}(i)\right\}\right), h_i^{(l,s)} = \text{COMBINE}\left(h_i^{(l-1,s)}, h_i^{'(l,s)}\right), \tag{5}$$

where $\mathcal{N}^{(s)}(i)$ denotes the neighbors of node i at s-th scale. After propagating messages within each scale, we employ an attention mechanism to fuse the features across scales.

$$h_i^{(l)} = \sum_{s=1}^S \alpha_i^{(s)} h_i^{(l,s)}, \quad \alpha_i^{(s)} = (W_{query} h_i^{(l,s)}) \star (W_{key} h_i^{(l-1)}), \tag{6}$$

where $\alpha_i^{(s)}$ denotes the attention weight, W_{query} and W_{key} are learnable parameters, and $\star$ denotes the cosine similarity computation. Building on this hierarchical representation, we parametrize a coupled Graph ODE to model the continuous-time evolution of it's latent state.

$$\frac{dh_i^t}{dt} = \Phi\left(h_1^t, h_2^t, \cdots, h_N^t, \mathrm{G}, \Theta\right). \tag{7}$$

Here, Φ denotes the multi-scale message passing function, G represents the hierarchical graph structure, and Θ encapsulates the model parameters. Given the ODE function Φ, the node initial values, and the corresponding contextual vector, the latent dynamics can be solved by any ODE solver like Runge-Kutta [37].

$$h_i^{t_1} \cdots h_i^{t_T} = \text{ODESolve}(\Phi, [h_1^0, h_2^0 \cdots h_N^0]). \tag{8}$$

This formulation allows the model to theoretically obtain the hidden state at any arbitrary timestep, provided that the Neural ODE fits the underlying dynamics well.

Neural Auto-correction. Although the multi-scale Graph ODE framework provides a global continuous-time evolution, it still relies on the strong assumption that a learnable ODE exists to model the concrete physical dynamics. Moreover, neural ODEs relying on numerical solvers may fail to capture the system's intrinsic nonlinear features, making it difficult to handle error divergence during evolution. To address this, we introduce a neural auto-correction module that imposes a discrete dynamical regime in the latent space. Thus we can capture complex corrections at selected time steps in a heuristic Markov chain manner.

At each correction step, a convolutional encoder first compresses the current latent representation into a compact state. Subsequently, after the discrete evolution block, a transposed convolution-based decoder reconstructs the refined latent field. This operation acts as a "Markov state observer" [8], learning a discrete single-step mapping in a more compact latent space. The formulation is as follows:

$$E(\cdot) = \sigma(\text{LN}(\text{Conv2d}(\cdot))), \quad D(\cdot) = \text{Tanh}(\text{UnConv2d}(\cdot)). \tag{9}$$

Further, between the encoder and decoder lies a neural block that embodies a discrete transition operator similar to a Markov kernel. Technically, it consists of a 1×1 *Conv2D*, which strips away extraneous channels to focus on the most critical latent features. Then, it is followed by parallel *GroupConv2D* [41] operators to process these features in multiple subspaces. We utilize $\mathcal{R}$ to represent the combination of the 1×1 *Conv2D* and the parallel *GroupConv2D*. Then the whole evolution can be formulated as:

$$z(k+1) \ = \ \mathcal{D}(\mathcal{R}\big(E(z(k))\big)), \tag{10}$$

where $z(k)$ denotes the latent embedding at discrete step k. Crucially, this transformation depends only on the current embedding $z(k)$, thus fulfilling a Markov property at each correction step. By encapsulating key spatiotemporal features in $z(k)$ and evolving them into $z(k+1)$ via $\mathcal{R}(\cdot)$, the neural auto-correction module adaptively regularizes the global continuous-time solution. On one hand, it provides a local perspective on how errors can be contained and corrected in each interval. On the other hand, it retains a sufficient memory of the system's evolving states without requiring full historical trajectories.

Inference Strategy. During inference, we employ an iterative strategy that combines the multi-scale Graph ODE module and the neural auto-correction module to generate continuous accurate predictions. Starting from the initial state $h_i(t_0)$, the model proceeds as follows:

$\star$ *Continuous Evolution*: The multi-scale Graph ODE module evolves the latent states continuously from t_k to t_{k+1}:

$$h_i(t_{k+1}) = h_i(t_k) + \int_{t_k}^{t_{k+1}} f_\theta\left(h_i(t), \{h_j(t)\}_{j \in N(i)}, t\right) dt. \tag{11}$$

$\star$ *Discrete Correction*: At each discrete time step t_{k+1}, the neural auto-correction module refines the predicted state:

$$h_i^\omega(t_{k+1}) = h_i(t_{k+1}) + \lambda r_\psi\left(h_i(t_k)\right), \tag{12}$$

where r_ψ denotes the neural auto-correction module, and λ is the hyperparameter for balance. The corrected state $h_i^\omega(t_{k+1})$ is used as the initial state for the next time step, and the process repeats until the final prediction time is reached. This inference strategy ensures that the model can generate accurate and stable predictions over long time horizons, while maintaining the continuity and smoothness of the learned dynamics.

3.3 Decoder

Upon obtaining the latent states on the customized grids, we project them back to the original continuous spatial domain for final predictions. Specifically, for a query coordinate $q_m = \{x_m, y_m\}$, we first identify the corresponding grid cell v_m and establish connections to its four vertices, denoted $\{v_{m1}, v_{m2}, v_{m3}, v_{m4}\}$. We then initialize the representation of q_m with a single step Gabor filter transformation $h_m = g_1(q_m)$, and then perform a message-passing update just as in Eq (4). Further, we can obtain the corresponding predictions through a 2-layer MLP decoder $\mathcal{D}(\cdot)$.

3.4 Theoretical Analysis

Theorem 3.1 (***Error Bounding via Hybrid Continuous-Discrete Latent Corrections***)**.** *Consider a spatiotemporal dynamical system whose latent representation $y(t)$ is intended to follow an ideal trajectory $y^*(t)$. The learned dynamics are modeled by a Neural ODE:*

$$\frac{d}{dt}y(t) = \mathcal{F}(y(t), \mathcal{G}, \Theta), \tag{13}$$

with initial condition $y(t_0)$. The function $\mathcal{F}$, representing the multi-scale graph message passing, is assumed to be $L_\mathcal{F}$-Lipschitz continuous with respect to $y(t)$. Periodic corrections are applied at discrete time steps $t_k = t_0 + k \cdot \Delta t_{\mathrm{corr}}$. Let $y(t_k^-)$ be the state before correction and $y(t_k^+)$ be the state after applying a neural auto-correction operator $\mathcal{C}_\psi$:

$$y(t_k^+) = y(t_k^-) + \mathcal{C}_\psi(y(t_k^-)). \tag{14}$$

The error is defined as $e(t) = \|y^*(t) - y(t)\|$. We assume:

- *(a) The combined error from the ODE model discrepancy (vis-à-vis $y^*(t)$) and numerical solver inaccuracies over one interval Δt_{corr} is bounded by $\mathcal{E}_{ODE}$, such that $e(t_{k+1}^-) \le e^{L_\mathcal{F}\Delta t_{\mathrm{corr}}} e(t_k^+) + \mathcal{E}_{ODE}$.*
- *(b) The correction operator $\mathcal{C}_\psi$ reduces the error proportionally and introduces a bounded residual, i.e., $e(t_k^+) \le \kappa \cdot e(t_k^-) + \delta_\mathcal{C}$, where $0 \le \kappa < 1$ is a contraction coefficient and $\delta_\mathcal{C} \ge 0$ is a base error from the corrector.*

Then, for $K = \lfloor (t - t_0)/\Delta t_{\mathrm{corr}} \rfloor$ correction steps, if the effective error amplification per cycle $\alpha_{\mathrm{eff}} = \kappa e^{L_\mathcal{F}\Delta t_{\mathrm{corr}}} < 1$, the error at time $t \approx t_0 + K\Delta t_{\mathrm{corr}}$ is bounded by:

$$\|e(t)\| \le \alpha_{\mathrm{eff}}^K \left(e(t_0) - \frac{\kappa\mathcal{E}_{ODE} + \delta_\mathcal{C}}{1 - \alpha_{\mathrm{eff}}} \right) + \frac{\kappa\mathcal{E}_{ODE} + \delta_\mathcal{C}}{1 - \alpha_{\mathrm{eff}}}. \tag{15}$$

This can be expressed more generally as:

$$\|e(t)\| \le C_1 \cdot \alpha_{\mathrm{eff}}^{\lfloor (t-t_0)/\Delta t_{\mathrm{corr}} \rfloor} + C_2, \tag{16}$$

where $C_1 = \max(0, e(t_0) - C_2)$ or simply $e(t_0)$ for a looser bound, $C_2 = \frac{\kappa\mathcal{E}_{ODE} + \delta_\mathcal{C}}{1 - \alpha_{\mathrm{eff}}}$ is the asymptotic error floor, and $\alpha_{\mathrm{eff}} \in [0, 1)$ quantifies the error decay rate per correction cycle, dependent on the system's Lipschitz constant, the correction interval, and the corrector's efficacy. This bound demonstrates that the error converges to a non-zero floor C_2 if $\alpha_{\mathrm{eff}} < 1$.

4 Experiment

4.1 Experimental Settings

Datasets. To comprehensively illustrate the property and efficacy of the extrapolations obtained from `CoPS`, we conduct experiments on diverse synthetic and real-world datasets. ✤ **For synthetic datasets**, we first choose *Navier-Stokes* [29] and *Rayleigh–Bénard Convection* [43], which are directly generated by numeric PDE solvers. Then, we select *Prometheus* [48], which is a large-scale combustion dataset simulated with industrial software. ✤ **For real-world datasets**, we choose *WeatherBench* [36], a dataset for weather forecasting and climate modeling. We also select *Kuroshio* [49], which provides vector data of sea surface stream velocity from the Copernicus Marine Service. See Appendix A for detailed descriptions of all these datasets.

Baselines. We evaluate our model against three baselines representing the state-of-the-art in continuous modeling. **MAgNet** [5] employs an "`Encode-Interpolate-Forecast`" scheme. Specifically, MAgNet employs the nearest neighbor interpolation technique to generalize to new query points.

Subsequently, it forecasts the evolutionary trends by leveraging a GNN-based message passing neural network. **DINO** [52] learns PDE's flow to forecast its dynamical evolution by leveraging a spatial implicit neural representation modulated by a context vector and modeling continuous-time evolution with a learned latent ODE. This is the closest approach to our method. **ContiPDE** [39] formulates the task as a double observation problem. It utilizes recurrent GNNs to roll out predictions of anchor states from the IC, and employs spatiotemporal attention observer to estimate the state at the query position from these anchor states. We utilize the official implementation for all models and tune their experimental settings to follow the requirements of our tasks.

Tasks. We assess the performance of `CoPS` across diverse forecasting tasks to evaluate their efficacy in various scenarios. We use the mean squared error (MSE) as the performance measurement. ***(i)*** **Sparse Flexibility** - Gauging the effectiveness of predicting global field evolution based on observations with diverse degrees of sparsity (subsampling ratio of 25%, 50%, 75%). ***(ii)*** **Time Flexibility** - Evaluating our model's performance in extrapolating beyond the training horizon. Here we design two experimental setups: long-term extrapolation beyond the training horizon, and continuous-time prediction for intermediate points between discrete time steps. ***(iii)*** **Resolution Generalization** - Investigating the performance of generalizing to new resolution (up or down). ***(iv)*** **Super-resolution** - Investigating the model's effectiveness of super-resolution query and extrapolation. ***(v)*** **Noise Robustness** - Exploring the robustness of our model against varying noise ratios. ***(vi)*** **Ablation Study** - Investigating contributes of each key component to the performance of `CoPS`.

Implementation. For synthetic data like Navier-Stokes, the train and test sets differ only by their initial conditions. The samples are partitioned in a 7:2:1 ratio into training, validation, and test sets. For real-world data like WeatherBench, we use the historical ERA5 global atmospheric reanalysis data. The data was partitioned strictly by date to prevent any data leakage from the future into the training process. Specifically, the train set uses data from 1979-2018, the validation set from 2019, and the test set from 2020-2022. To ensure fairness, we use partial observations of the initial state as input and supervise the training with future real observations at discrete time steps (only using observed future values). In evaluating spatial continuity, we perform subsampling of the full initial state at rates of {25%, 50%, and 75%}, assessing both observed and unobserved points. For temporal continuity, we evaluate three criteria based on different subsampling rates: predictions within the training horizon, long-term predictions beyond the training horizon, and continuous-time predictions for intermediate points between discrete time steps. More details are illustrated in Appendix C.

4.2 Main Results

Space Flexibility. To evaluate the spatial querying ability of our method, we adjust the number of available measurement points by using different proportions of observation points (25%, 50%, 75%) in the training data. We report the MSE compared to the ground truth on four datasets, as shown in Table 1. From the table, we see that our method (Ours) significantly outperforms the baseline methods on all datasets and at all observation sparsity levels. On the Navier-Stokes dataset, with only 25% of the observation data, our method achieves MSE of 2.964E-03 and 5.743E-03 under the In-s and Ext-s settings, which are much better than ContiPDE (5.456E-03 and 9.523E-03) and DINo (1.074E-02 and 2.537E-02). As the observation ratio increases to 50% and 75%, our model's performance further improves. The MSE for In-s decrease to 2.253E-03 and 1.464E-03, and for Ext-s decrease to 4.571E-03 and 2.872E-03. On the Kuroshio dataset, even with only 25% observation data, our method achieves MSE of 1.285E-03 and 2.381E-03 under In-s and Ext-s, significantly better than other methods. For example, ContiPDE's MSE are 2.352E-03 and 3.763E-03, and DINo's are 3.737E-03 and 5.923E-03. As the observation ratio increases, our method's MSE further decreases, and its accuracy becomes higher. On the Prometheus and WeatherBench datasets, our method also performs excellently. Especially on the WeatherBench dataset, when the observation ratio is 75%, our method achieves MSE of 3.585E-03 and 8.472E-03 under In-s and Ext-s, much lower than other baseline methods. These results show that our method effectively recovers the global scene from partial observations on different datasets and observation sparsity levels. It demonstrates robustness in handling sparse data and spatial generalization.

Time Flexibility. To evaluate our method's performance in temporal extrapolation, we train and evaluate the model with observation subsampling ratio of 50% and 100%. We use three extrapolation modes: (1) Extrapolation within the discrete training range (In-t); (2) Extrapolation beyond the discrete training range (Ext-t); (3) Extrapolation at intermediate points between time steps (Con-t).

Table 1: To evaluate the spatial inquiry power of our method, we vary the number of available measurement points in the data for training from 25%, 50% and 75% amount of observations. We report MSE compared to the ground truth solution.

Model		Navier-Stokes			Kuroshio			Prometheus			WeatherBench		
		s=25%	s=50%	s=75%	s=25%	s=50%	s=75%	s=25%	s=50%	s=75%	s=25%	s=50%	s=75%
MAgNet	In-s	3.164E-02	2.775E-02	2.268E-02	7.104E-03	6.523E-03	4.845E-03	1.694E-02	1.273E-02	1.186E-02	3.635E-02	3.028E-02	2.483E-02
	Ext-s	5.846E-02	4.285E-02	3.114E-02	9.464E-03	8.173E-03	7.518E-03	2.775E-02	2.322E-02	1.724E-02	5.356E-02	4.972E-02	3.295E-02
DINo	In-s	1.074E-02	9.564E-03	7.554E-03	3.737E-03	3.332E-03	2.884E-03	1.223E-02	8.005E-03	5.657E-03	2.365E-02	2.186E-02	1.922E-02
	Ext-s	2.537E-02	1.764E-02	9.248E-03	5.923E-03	5.271E-03	4.487E-03	1.784E-02	1.246E-02	8.324E-03	3.823E-02	2.746E-02	2.588E-02
ContiPDE	In-s	5.456E-03	4.176E-03	3.825E-03	2.352E-03	1.947E-03	1.503E-03	8.462E-03	6.084E-03	4.763E-03	1.542E-02	1.294E-02	1.056E-02
	Ext-s	9.523E-03	7.975E-03	6.231E-03	3.763E-03	2.531E-03	2.084E-03	1.125E-02	8.355E-03	6.674E-03	2.172E-02	1.653E-02	1.474E-02
Ours	In-s	**2.964E-03**	**2.253E-03**	**1.464E-03**	**1.285E-03**	**7.253E-04**	**5.253E-04**	**5.176E-03**	**3.722E-03**	**2.746E-03**	**8.766E-03**	**5.249E-03**	**3.585E-03**
	Ext-s	**5.743E-03**	**4.571E-03**	**2.872E-03**	**2.381E-03**	**1.577E-03**	**9.272E-04**	**8.264E-03**	**5.142E-03**	**4.287E-03**	**1.769E-02**	**1.056E-02**	**8.472E-03**

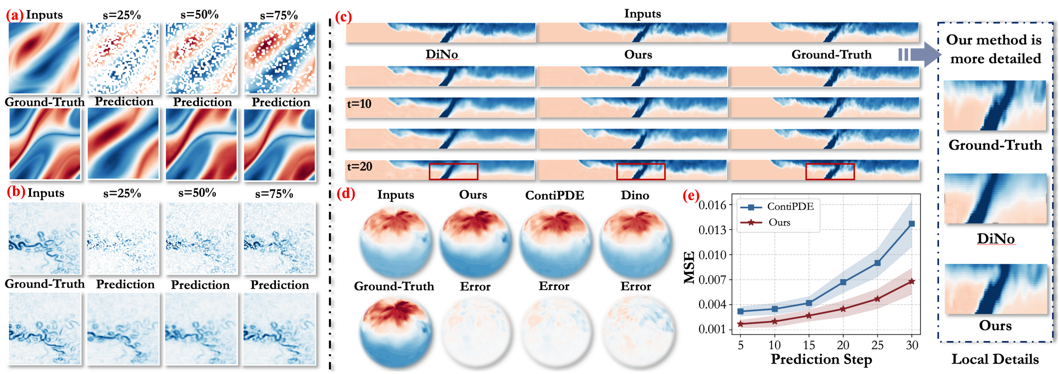

Figure 3: Figure(a) and (b) shows the perdiction performance based on observations with **diverse ratio of subsampling** (25%, 50%, 75%) on the Navier-Stokes and Kuroshio dataset. Figure(c) demonstrates **long-term extrapolation beyond the training horizon** on the Prometheus dataset, and Figure(e) shows the evolution of prediction errors over time steps, revealing the increasing error with longer prediction steps. Figure(d) illustrates **continuous-time prediction** for intermediate points between discrete time steps on the WeatherBench dataset.

We report the mean squared error (MSE) on the Navier-Stokes, Rayleigh–Bénard Convection, and Prometheus datasets, as shown in Table 2. From Table 2, we see that our method achieves the best performance on mostly all datasets and extrapolation settings. For example, on the Navier-Stokes dataset with a subsampling ratio of 50%, our method achieves MSE of 2.832E-03 (In-t), 5.764E-03 (Ext-t), and 4.135E-03 (Con-t). These are significantly better than ContiPDE (5.054E-03, 8.342E-03, 6.447E-03) and DINo (7.651E-03, 1.074E-02, 9.971E-03). When the subsampling ratio increases to 100%, the performance improves further. Our method's MSE decrease to 1.132E-03 (In-t), 2.364E-03 (Ext-t), and 1.735E-03 (Con-t). On the Rayleigh–Bénard Convection dataset, our method also achieves MSE of 6.522E-04 (In-t) and 1.327E-03 (Ext-t), significantly better than other baseline methods. This shows that our method can make accurate predictions within the discrete training range. It can also effectively extrapolate to time points beyond the training range. When we perform continuous-time prediction at intermediate points between time steps (Con-t), our method also performs excellently. It achieves the smallest prediction error. Overall, these results demonstrate the strong capability of our method in continuous-time modeling. It provides accurate and reliable predictions under different temporal extrapolation settings.

Visualization and Analysis. Figure 3 provides compelling visual evidence for the superior performance of our proposed `CoPS` across a range of challenging tasks. Specifically, Figure 3 (a) and (b) demonstrate that `CoPS` can effectively model the entire physical field and learn the evolution trends of unobserved points only with partial observations. When the subsampling ratio reaches 75%, our model can achieve a precise prediction on Navier-Stokes and Kuroshio datasets. As observed in Figure 3 (c) and (e), our method significantly outperforms DINO and ContiPDE in long-term forecasting on the Prometheus dataset by effectively suppressing error accumulation, thereby maintaining high-fidelity details where baselines exhibit blurring. This robust temporal modeling is further evidenced in Figure 3 (d), where `CoPS` accurately predicts intermediate states between discrete time steps. This showcases its ability to perform true, dynamically consistent integration rather than simple interpolation, confirming its superior continuous-time modeling capabilities.

Table 2: We evaluate the temporal extrapolation performance of our method. Models are trained and evaluated with observation subsampling ratio of 50% and 100%. We employ three kinds of extrapolation, containing: (1) extrapolation within discrete training horizon (In-t); (2) extrapolation exceeding discrete training horizon (Ext-t); and (3) extrapolation of intermediate points between time steps (Con-t). We report MSE compared to the ground truth solution.

	Navier-Stokes			Rayleigh–Bénard Convection			Prometheus		
	In-t	Ext-t	Con-t	In-t	Ext-t	Con-t	In-t	Ext-t	Con-t
				$s = 50\%$ subsampling ratio					
MAgNet	1.253E-02	2.601E-02	–	9.421E-03	1.335E-02	–	1.138E-02	1.824E-02	–
DINo	7.651E-03	1.074E-02	9.971E-03	5.748E-03	8.472E-03	7.342E-03	8.321E-03	1.117E-02	8.852E-03
ContiPDE	5.054E-03	8.342E-03	6.447E-03	3.744E-03	3.814E-03	2.908E-03	5.435E-03	8.154E-03	6.637E-03
Ours	**2.832E-03**	**5.764E-03**	**4.135E-03**	**1.526E-03**	**2.328E-03**	**1.765E-03**	**3.374E-03**	**6.678E-03**	**5.355E-03**
				$s = 100\%$ subsampling ratio					
MAgNet	7.429E-03	1.273E-02	–	4.837E-03	7.235E-03	–	7.938E-03	1.024E-02	–
DINo	4.352E-03	7.374E-03	5.271E-03	2.248E-03	4.872E-03	3.742E-03	5.721E-03	8.217E-03	6.288E-03
ContiPDE	2.655E-03	4.742E-03	3.847E-03	1.244E-03	3.214E-03	2.308E-03	3.835E-03	6.554E-03	4.037E-03
Ours	**1.132E-03**	**2.364E-03**	**1.735E-03**	**6.522E-04**	**1.327E-03**	**1.061E-03**	**2.877E-03**	**5.174E-03**	**3.852E-03**

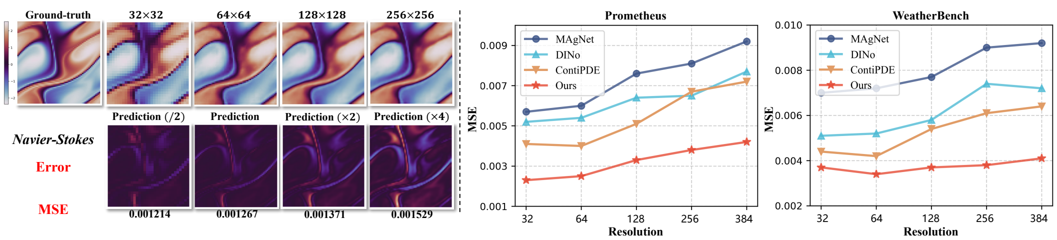

Figure 4: *Left* shows the inference performance for super-resolution on Navier-Stokes dataset. *Right* demonstrates the resolution generalization capability compared with MAgNet, DINo, and ContiPDE.

4.3 Model Analysis

Analysis of Super-resolution. In the super-resolution experiment presented in Figure 4 (*left*), the results illustrate the inference performance on the Navier-Stokes dataset across resolutions of 32×32, 64×64, 128×128, and 256×256. The results consistently demonstrate improved prediction quality as resolution increases. Compared to baseline models (MAgNet, DINo, and ContiPDE), the proposed model effectively reduces prediction error at high resolutions, maintaining a lower MSE and providing clearer, more accurate predictions of the original Navier-Stokes patterns.

Resolution Generalization. As observed in Figure 4 (*right*), the folding diagrams depict the resolution generalization capabilities for both Prometheus and WeatherBench datasets. The proposed CoPS consistently outperforms the comparative models (MAgNet, DINo, ContiPDE) at all resolution levels, achieving the lowest MSE value. This result highlights CoPS's robustness and effectiveness in handling resolution variations. The findings emphasize the model's ability to generalize across different resolutions, which is crucial for applications requiring detailed and precise predictions in complex dynamic systems.

Table 3: Ablation study on Kuroshio. (Report MSE)

	In-s / In-t	Ext-s / In-t	In-s / Ext-t	Ext-s / Ext-t
w/o MFN	1.027E-03	1.946E-03	2.458E-03	3.149E-03
w/o MGO	1.594E-03	2.357E-03	3.053E-03	4.175E-03
w/o NAC	1.428E-03	2.216E-03	2.742E-03	3.657E-03
Ours	**7.253E-04**	**1.577E-03**	**2.161E-03**	**2.938E-03**

Ablation Study. To evaluate the contribution and importance of each component in the proposed CoPS, we conduct corresponding ablation experiments, and use MSE error as metric. Our model variants are as follows: ❶ **CoPS w/o MFN**, we remove the multiplicative filter network and use an MLP. ❷ **CoPS w/o MGO**, we remove the multi-scale Graph ODE module. ❸ **CoPS w/o NAC**, we remove the neural auto-correction module. Table 3 shows the comprehensive results of our ablation experiment. The results of the ablation experiments show that removing any component results in a decrease in predictive performance, further proving the critical role of these components in CoPS framework. Specifically, the performance declines after removing the multi-scale Graph ODE module is particularly evident, proving its importance in capturing internal dynamic representations.

5 Conclusion

In this paper, we introduce CoPS, a novel data-driven framework for modeling continuous spatiotemporal dynamics from partial observations. To overcome discretization, we first customize geometric grids and employ multiplicative filter network to fuse and encode spatial information with the corresponding observations. After encoding to the concrete grids, CoPS model continuous-time dynamics by designing multi-scale graph ODEs, while introducing a Marcov-based neural auto-correction module to assist and constrain the continuous extrapolations. Extensive experiments on both synthetic and real-world datasets demonstrate the superior performance of CoPS, which underlines the potential to provide a robust method for accurate long-term predictions in face of sparse and unstructured data.

Acknowledgement

This work was partially supported by the NSFC Program (No. 62202042, No. 6212780016); in part by the Supported by the Fundamental Research Funds for the Central Universities (No. 2024JBMC031); in part by the Aeronautical Science Foundation of China (No. ASFC-2024Z0710M5002; in part by the by the OpenFund of Advanced Cryptography and System Security Key Laboratory of Sichuan Province(No. SKLACSS-202312); the Guangdong S&T Programme (No. 2024B0101030002), the Ministry of Industry and Information Technology of China, the National Key Research and Development Program of China (No. 2023YFB3307500), and the Science and Technology Innovation Project of Hunan Province (No. 2023RC4014).

References

[1] Md Ashiqur Rahman, Zachary E Ross, and Kamyar Azizzadenesheli. U-no: U-shaped neural operators. *arXiv e-prints*, pages arXiv–2204, 2022.

[2] Timothy Barth, Raphaèle Herbin, and Mario Ohlberger. Finite volume methods: foundation and analysis. *Encyclopedia of computational mechanics second edition*, pages 1–60, 2018.

[3] Marin Biloš, Johanna Sommer, Syama Sundar Rangapuram, Tim Januschowski, and Stephan Günnemann. Neural flows: Efficient alternative to neural odes. *Advances in neural information processing systems*, 34:21325–21337, 2021.

[4] Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, Karthik Kashinath, and Anima Anandkumar. Spherical fourier neural operators: Learning stable dynamics on the sphere. In *International conference on machine learning*, pages 2806–2823. PMLR, 2023.

[5] Oussama Boussif, Yoshua Bengio, Loubna Benabbou, and Dan Assouline. Magnet: Mesh agnostic neural pde solver. *Advances in Neural Information Processing Systems*, 35:31972–31985, 2022.

[6] Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6989–6997, 2023.

[7] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[8] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

[9] Michaël Defferrard, Martino Milani, Frédérick Gusset, and Nathanaël Perraudin. Deepsphere: a graph-based spherical cnn. In *International Conference on Learning Representations*, 2020.

[10] Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J Zico Kolter. Multiplicative filter networks. In *International Conference on Learning Representations*, 2020.

[11] Zaige Fei, Fan Xu, Junyuan Mao, Yuxuan Liang, Qingsong Wen, Kun Wang, Hao Wu, and Yang Wang. Open-CK: A large multi-physics fields coupling benchmarks in combustion kinetics. In *The Thirteenth International Conference on Learning Representations*, 2025.

[12] Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning visual predictive models of physics for playing billiards. *arXiv preprint arXiv:1511.07404*, 2015.

[13] RH Gallagher, JT Oden, C Taylor, and OC Zienkiewicz. Finite element. *Analysis: Fundamentals. Prentice Hall, Englewood Cliffs*, 1975.

[14] Yuan Gao, Hao Wu, Ruiqi Shu, Huanshuo Dong, Fan Xu, Rui Ray Chen, Yibo Yan, Qingsong Wen, Xuming Hu, Kun Wang, et al. Oneforecast: a universal framework for global and regional weather forecasting. *arXiv preprint arXiv:2502.00338*, 2025.

[15] Zhihan Gao, Xingjian Shi, Hao Wang, Yi Zhu, Yuyang Bernie Wang, Mu Li, and Dit-Yan Yeung. Earthformer: Exploring space-time transformers for earth system forecasting. *Advances in Neural Information Processing Systems*, 35:25390–25403, 2022.

[16] Nicholas Geneva and Nicholas Zabaras. Modeling the dynamics of pde systems with physics-constrained deep auto-regressive networks. *Journal of Computational Physics*, 403:109056, 2020.

[17] Daniele Grattarola and Pierre Vandergheynst. Generalised implicit neural representations. *Advances in Neural Information Processing Systems*, 35:30446–30458, 2022.

[18] Zhenyun Hao, Jianing Hao, Zhaohui Peng, Senzhang Wang, Philip S Yu, Xue Wang, and Jian Wang. Dy-hien: Dynamic evolution based deep hierarchical intention network for membership prediction. In *WSDM*, pages 363–371, 2022.

[19] Valerii Iakovlev, Markus Heinonen, and Harri Lähdesmäki. Learning continuous-time pdes from sparse data with graph neural networks. *arXiv preprint arXiv:2006.08956*, 2020.

[20] Valerii Iakovlev, Markus Heinonen, and Harri Lähdesmäki. Learning space-time continuous latent neural pdes from partially observed states. *Advances in Neural Information Processing Systems*, 36, 2024.

[21] Ehsan Kharazmi, Zhongqiang Zhang, and George Em Karniadakis. Variational physics-informed neural networks for solving partial differential equations. *arXiv preprint arXiv:1912.00873*, 2019.

[22] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[23] David Knigge, David Wessels, Riccardo Valperga, Samuele Papa, Jan-Jakob Sonke, Erik Bekkers, and Efstratios Gavves. Space-time continuous pde forecasting using equivariant neural fields. *Advances in Neural Information Processing Systems*, 37:76553–76577, 2024.

[24] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

[25] Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems*, 34:26548–26560, 2021.

[26] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

[27] Jun Li, Gan Sun, Guoshuai Zhao, and H Lehman Li-wei. Robust low-rank discovery of data-driven partial differential equations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 767–774, 2020.

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

[29] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[30] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020.

[31] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[32] Xihaier Luo, Wei Xu, Yihui Ren, Shinjae Yoo, and Balu Nadiga. Continuous field reconstruction from sparse observations with implicit neural networks. *arXiv preprint arXiv:2401.11611*, 2024.

[33] Mohammed S Mechee and Sameeah H Aidi. Generalized euler and runge-kutta methods for solving classes of fractional ordinary differential equations. *International Journal of Nonlinear Analysis and Applications*, 13(1):1737–1745, 2022.

[34] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.

[35] Bogdan Raonic, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. Convolutional neural operators for robust and accurate learning of pdes. *Advances in Neural Information Processing Systems*, 36, 2024.

[36] Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russel, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, et al. Weatherbench 2: A benchmark for the next generation of data-driven global weather models. *arXiv preprint arXiv:2308.15560*, 2023.

[37] Michael Schober, David K Duvenaud, and Philipp Hennig. Probabilistic ode solvers with runge-kutta means. *Advances in neural information processing systems*, 27, 2014.

[38] Suihong Song, Tapan Mukerji, and Jiagen Hou. Bridging the gap between geophysics and geology with generative adversarial networks. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11, 2021.

[39] Janny Steeven, Nadri Madiha, Digne Julie, and Wolf Christian. Space and time continuous physics simulation from partial observations. In *ICLR*, 2024.

[40] Rob Stoll, Jeremy A Gibbs, Scott T Salesky, William Anderson, and Marc Calaf. Large-eddy simulation of the atmospheric boundary layer. *Boundary-Layer Meteorology*, 177:541–581, 2020.

[41] Cheng Tan, Zhangyang Gao, Lirong Wu, Yongjie Xu, Jun Xia, Siyuan Li, and Stan Z Li. Temporal attention unit: Towards efficient spatiotemporal predictive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18770–18782, 2023.

[42] Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. In *ICLR*, 2023.

[43] Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. Towards physics-informed deep learning for turbulent flow prediction. *KDD*, 2019.

[44] Sifan Wang, Xinling Yu, and Paris Perdikaris. When and why pinns fail to train: A neural tangent kernel perspective. *J. Comput. Phys.*, 2020.

[45] Weiyan Wang, Xingjian Shi, Ruiqi Shu, Yuan Gao, Rui Ray Chen, Kun Wang, Fan Xu, Jinbao Xue, Shuaipeng Li, Yangyu Tao, et al. Beamvq: Beam search with vector quantization to mitigate data scarcity in physical spatiotemporal forecasting. *arXiv preprint arXiv:2502.18925*, 2025.

[46] Hao Wu, Yuan Gao, Ruiqi Shu, Zean Han, Fan Xu, Zhihong Zhu, Qingsong Wen, Xian Wu, Kun Wang, and Xiaomeng Huang. Turb-l1: Achieving long-term turbulence tracing by tackling spectral bias. *arXiv preprint arXiv:2505.19038*, 2025.

[47] Hao Wu, Changhu Wang, Fan Xu, Jinbao Xue, Chong Chen, Xian-Sheng Hua, and Xiao Luo. Pure: Prompt evolution with graph ode for out-of-distribution fluid dynamics modeling. *Advances in Neural Information Processing Systems*, 37:104965–104994, 2024.

[48] Hao Wu, Huiyuan Wang, Kun Wang, Weiyan Wang, Changan Ye, Yangyu Tao, Chong Chen, Xian-Sheng Hua, and Xiao Luo. Prometheus: Out-of-distribution fluid dynamics modeling with disentangled graph ode. In *Proceedings of the 41st International Conference on Machine Learning*, page PMLR 235, Vienna, Austria, 2024. PMLR.

[49] Hao Wu, Kangyu Weng, Shuyi Zhou, Xiaomeng Huang, and Wei Xiong. Neural manifold operators for learning the evolution of physical dynamics. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3356–3366, 2024.

[50] Hao Wu, Fan Xu, Chong Chen, Xian-Sheng Hua, Xiao Luo, and Haixin Wang. Pastnet: Introducing physical inductive biases for spatio-temporal video prediction. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 2917–2926, 2024.

[51] Fan Xu, Nan Wang, Hao Wu, Xuezhi Wen, Xibin Zhao, and Hai Wan. Revisiting graph-based fraud detection in sight of heterophily and spectrum. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 9214–9222, 2024.

[52] Yuan Yin, Matthieu Kirchmeyer, Jean-Yves Franceschi, Alain Rakotomamonjy, and Patrick Gallinari. Continuous pde dynamics forecasting with implicit neural representations. *arXiv preprint arXiv:2209.14855*, 2022.

[53] Yanfu Zhang, Shangqian Gao, Jian Pei, and Heng Huang. Improving social network embedding via new second-order continuous graph neural networks. In *KDD*, pages 2515–2523, 2022.

[54] Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Zhou Qin, and Wenwu Zhu. Dynamic graph neural networks under spatio-temporal distribution shift. *Advances in neural information processing systems*, 35:6074–6089, 2022.

[55] Xi Zhao, Linghsuan Meng, Xu Tong, Xiaotong Xu, Wenxin Wang, Zhongrong Miao, Dapeng Mo, et al. A novel computational fluid dynamic method and validation for assessing distal cerebrovascular microcirculatory resistance. *Computer Methods and Programs in Biomedicine*, 230:107338, 2023.

[56] David Zheng, Vinson Luo, Jiajun Wu, and Joshua B Tenenbaum. Unsupervised learning of latent physical properties using perception-prediction networks. *arXiv preprint arXiv:1807.09244*, 2018.

[57] Yanping Zheng, Hanzhi Wang, Zhewei Wei, Jiajun Liu, and Sibo Wang. Instant graph neural networks for dynamic graphs. In *KDD*, pages 2605–2615, 2022.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the contributions regarding the CoPS framework.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We thoroughly discuss the limitations of our work and propose potential directions for future research in Appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The proofs of corresponding theorems are provided in Appendix.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed workflow and experimental model settings in Appendix.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: In the abstract, we provide URL link to both the source code.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide experimental settings in Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The experiments in this paper report variance measurements.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide information on the computer resources in Experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: All aspects of this work comply with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed broader impacts in Appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited all used public datasets and compared baselines.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This study did not involve human participants, so no risks, disclosures, or IRB approvals were required or obtained.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core ideas proposed in this paper were developed without any involvement of large language models.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Detailed Description of Datasets

Navier-Stokes Equations. Navier-Stokes Equations [29] describe the motion of fluid substances such as liquids and gases. These equations are a set of partial differential equations that predict weather, ocean currents, water flow in a pipe, and air flow around a wing, among other phenomena. The equations arise from applying Newton's second law to fluid motion, together with the assumption that the fluid stress is the sum of a diffusing viscous term proportional to the gradient of velocity, and a pressure term. The equations are expressed as follows:

$$\begin{aligned} \rho\left(\frac{\partial u}{\partial t} + u \cdot \nabla u\right) &= -\nabla p + \nabla \cdot \tau + f, \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) &= 0, \\ \frac{\partial (\rho E)}{\partial t} + \nabla \cdot ((\rho E + p)u) &= \nabla \cdot (\tau \cdot u) + \nabla \cdot (k \nabla T) + \rho f \cdot u, \end{aligned} \tag{17}$$

where u denotes the velocity field, ρ represents the density of the fluid, p is the pressure, τ is the viscous stress tensor, given by $\mu(\nabla u + (\nabla u)^T) - \frac{2}{3}\mu(\nabla \cdot u)\mathbf{I}$. E is the total energy per unit mass, $E = e + \frac{1}{2}|u|^2$, e is the internal energy per unit mass, T denotes the temperature, and k represents the thermal conductivity. All simulations were generated from the Navier-Stokes equation with a constant Reynolds number of 1e-5. We utilized a total of 1200 independent simulation samples.

Rayleigh-Bénard Convection. Rayleigh-Bénard Convection [43] is generated using the Lattice Boltzmann Method to solve the 2-dimensional fluid thermodynamics equations for two-dimensional turbulent flow. The general form of the equations is expressed as:

$$\begin{aligned} \nabla \cdot u &= 0, \\ \frac{\partial u}{\partial t} + (u \cdot \nabla)u &= -\frac{1}{\rho_0}\nabla p + \nu \nabla^2 u + [1 - \alpha(T - T_0)]X, \\ \frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla)T &= \kappa \nabla^2 T, \end{aligned} \tag{18}$$

where g is the gravitational acceleration, $\mathbf{X}$ is the acceleration due to the body-force of the fluid parcel, ρ_0 is the relative density, T represents temperature, T_0 is the average temperature, α denotes the coefficient of thermal expansion, and κ is the thermal conductivity coefficient. The simulation parameters for the dataset are as follows: Prandtl number = 0.71, Rayleigh number = 2.5×10^8, and the maximum Mach number = 0.1.

Prometheus. Prometheus [48] is a large-scale, out-of-distribution (OOD) fluid dynamics dataset designed for the development and benchmarking of machine learning models, particularly those that predict fluid dynamics under varying environmental conditions. This dataset includes simulations of tunnel and pool fires (representated as Prometheus-T and Prometheus-P in experiments), encompassing a wide range of fire dynamics scenarios modeled using fire dynamics simulators that solve the Navier-Stokes equations. Key features of the dataset include 25 different environmental settings with variations in parameters such as Heat Release Rate (HRR) and ventilation speeds. In total, the Prometheus dataset encompasses 4.8 TB of raw data, which is compressed to 340 GB. It not only enhances the research on fluid dynamics modeling but also aids in the development of models capable of handling complex, real-world scenarios in safety-critical applications like fire safety management and emergency response planning.

WeatherBench. WeatherBench [36] is a benchmark dataset designed for the evaluation and comparison of machine learning models in the context of medium-range weather forecasting. It is intended to facilitate the development of data-driven models that can improve weather prediction, particularly in the range from 1 to 14 days ahead. The dataset consists of historical weather data from multiple atmospheric variables, including temperature, pressure, humidity, wind speed, and geopotential height, at various global locations. The data is derived from the ERA5 reanalysis, which provides hourly estimates of the atmosphere's state at a resolution of 31 km for the period from 1950 to present. Its primary goal is to serve as a benchmarking tool to compare the performance of machine learning-based models against traditional numerical weather prediction methods. By focusing on data-driven techniques, it aims to push forward the development of models that can predict weather patterns in an interpretable and scalable manner, and thus contribute to improving operational weather forecasting systems.

Kuroshio. Kuroshio [49] is a dataset designed to study the dynamics of the Kuroshio Current, a major oceanic current in the Pacific Ocean that flows from the east coast of Taiwan and Japan, obtained from the Copernicus Marine Environment Monitoring Service (CMEMS). Key features of the dataset include high temporal and spatial resolution measurements, covering daily and monthly intervals, which allow for in-depth studies on the seasonal and inter-annual variability of the Kuroshio Current. It also includes data on associated oceanic phenomena such as eddy formation, upwelling, and interactions with surrounding currents like the Tsushima Current. This dataset covers the period from 1993 to 2024, and we use data from 1993–2020 for training, while data from 2021–2024 for validation and testing.

B Pseudocode of CoPS

Algorithm 1 Algorithm workflow of CoPS

Input: Initial observations $U_0 = \{(u(x_i, t_0), x_i)\}_{i=1}^{N_{obs}}$; Query set Q_{set}; Model parameters Θ_{all}; Hyperparameters $\Delta t_{corr}, \lambda_c$.
Output: Predictions $\hat{Y} = \{\hat{u}(t_q, x_q)\}$ for $(t_q, x_q) \in Q_{set}$.

/* **Stage 1: Initial Encoding and Grid Mapping** */
$\{h_i(t_0)\} \leftarrow \text{Encoder}_{\Theta_{enc}}(U_0)$ ▷ Encode observations to point embeddings ▷ Eq. 2, 3
$z(t_0)^+ \leftarrow \text{GridMapper}_{\Theta_{map}}(\{h_i(t_0)\})$ ▷ Map to initial latent grid state ▷ Eq. 4

/* **Stage 2: Iterative Latent Dynamics Prediction with Correction** */
$z_k^+ \leftarrow z(t_0)^+$; $t_k \leftarrow t_0$; $\mathcal{Z}_{traj} \leftarrow \{(t_0, z(t_0)^+)\}$
while $t_k < \max_{(t_q,\cdot)\in Q_{set}}\{t_q\}$ **do**
  $t_{next\_corr} \leftarrow t_k + \Delta t_{corr}$
  $z_{k+1}^- \leftarrow \text{ODESolve}(\Phi_{\Theta_{ode}}, z_k^+, [t_k, t_{next\_corr}])$ ▷ Continuous evolution ▷ Eq. 7, 8, 11
  Store evolution from z_k^+ to z_{k+1}^- in $\mathcal{Z}_{traj}$.
  $z_{k+1}^+ \leftarrow z_{k+1}^- + \lambda_c \cdot A_{\Theta_{ac}}(z_k^+)$ ▷ Apply discrete correction ▷ Eq. 9, 10, 12
  $z_k^+ \leftarrow z_{k+1}^+$; $t_k \leftarrow t_{next\_corr}$
end while

/* **Stage 3: Decoding to Query Locations** */
$\hat{Y} \leftarrow \{\}$
for each query (t_q, x_q) in Q_{set} **do**
  Identify grid cell v_{cell} and its vertices $\{v_j\}$ for x_q.
  $h_{q,init} \leftarrow \text{GaborFilter}(x_q)$ ▷ Initialize query point representation
  $h_{q,grid} \leftarrow \text{MessagePass}(\{z(t_q)_{v_j}\}, h_{q,init}, x_q, \{x_{v_j}\}; \Theta_{map})$ ▷ Refine using grid states ▷ Eq. 4 variant
  $\hat{u}(t_q, x_q) \leftarrow \text{MLPDecoder}(h_{q,grid}; \Theta_{dec})$
  Add $\hat{u}(t_q, x_q)$ to $\hat{Y}$.
end for

/* **Training:** Parameters $\Theta_{all} = \{\Theta_{enc}, \Theta_{map}, \Theta_{ode}, \Theta_{ac}, \Theta_{dec}\}$ are learned end-to-end by minimizing prediction error. */
return $\hat{Y}$

C Details of Implementation

To ensure fairness, we conducted all experiments on an NVIDIA-A100 GPU using the MSE loss over 200 epochs. We used Adam optimizer with a learning rate of 10^{-3} for training. The batch size was set to 16.

C.1 Model hyper-parameters

The core architecture of the model consists of three main modules: the spatial information encoder, the multi-scale graph ODE module, and the neural auto-regressive correction module. The spatial information encoder employs Multiplicative Filter Networks to fuse partial observations with spatial coordinate information and encode them into customized geometric grids. For regular arranged datasets, the customized grid is set to the general resolution. For irregular arranged datasets, it is uniformly set to 128×128. The MFN is configured with 5 layers and uses ReLU as the activation function to ensure efficient feature extraction. The hidden dimension is set to 128. Then the multi-scale graph ODE module utilizes a Runge-Kutta ode solver for numerical integration, with a time step size of 0.25 to ensure accurate modeling of temporal continuity. Further, the neural auto-regressive correction module performs corrections per integer time step. For this module, the Conv2d layer is downsampled to half the resolution, while the UpConv2d layer restores the grid to the original resolution. The parallel GroupConv2d operations are implemented with filter sizes of 3×3, 5×5, and 7×7. For inference, the correction weight λ is set to 0.5 to balance correction strength and model stability. Finally, in the decoder, we use a single step Gabor filter transformation to initial the features of query coordinates, and perform a 2-layers message-passing update to obtain the corresponding predictions.

C.2 Baseline implementation

MAgNet [5]. We utilize the official implementation of MAgNet, utilizing a graph neural network variant of the model. The configuration involves five message-passing steps. The architecture of all MLPs includes four layers, with each layer containing 128 neurons. Additionally, we set the dimensionality of the latent state at 128.

DINo [52]. We utilize the official implementation of DINo. Specifically, the encoder features an MLP, comprising three hidden layers with 512 neurons each, and Swish non-linearities. The dimension of each hidden layer is set to 100. Similarly, the dynamics function is realized through an MLP, which also includes three hidden layers, each containing 512 neurons and employing Swish activation function. The decoder is constructed with three layers, each with a capacity of 64 channels.

ContiPDE [39]. ContiPDE formulates the task as a double observation problem. It utilizes recurrent GNNs to roll out predictions of anchor states from the IC, and employs spatio-temporal attention observer to estimate the state at the query position from these anchor states. First, it utilize a two-layered MLP with 128 neurons, with Swish activation functions to encode features form sparse observations. Further, it uses a two-layered gated recurrent unit with a hidden vector of size 128, and a two-layered MLP with 128 neurons activated by the Swish function to realize the recurrent GNNs. Finally, it employs multi-head attention mechanism to decode and utilizes multi-head attention mechanism to realize continuous query.

D Broader impacts

The `CoPS` framework has broad positive impacts in several areas. First, it provides crucial technical support in scientific research, especially in geophysics, atmospheric science, and fluid dynamics. By performing continuous spatiotemporal simulations from sparse observational data, this method achieves high-precision predictions with limited sensor numbers, significantly improving the accuracy and timeliness of weather forecasts and disaster warnings.

E Limitations of This Study

While our proposed `CoPS` framework demonstrates significant advancements in modeling continuous spatiotemporal dynamics from partial observations, we acknowledge several limitations that provide avenues for future research. The first is the computational cost of multi-scale graph ODEs. It can be computationally intensive, especially for very fine-grained customized grids or a large number of scales. Future work could explore more efficient graph neural network architectures or adaptive scaling mechanisms to mitigate this. The second is the assumption of Markov property in auto-correction. While this simplifies the model and makes it computationally tractable, real-world physical systems might exhibit longer-range temporal dependencies that are not fully captured by this discrete correction mechanism.

F Proofs of Theorem 3.1

Let $e_k^- = e(t_k^-)$ denote the error just before the k-th correction (or at the start of the k-th ODE integration interval), and $e_k^+ = e(t_k^+)$ denote the error just after the k-th correction. The initial error is $e_0^- = e(t_0)$.

From assumption (b), after a correction at time t_k:

$$e_k^+ \leq \kappa e_k^- + \delta_{\mathcal{C}}. \tag{19}$$

From assumption (a), after ODE evolution from t_k to t_{k+1}:

$$e_{k+1}^- \leq e^{L_{\mathcal{F}} \Delta t_{\text{corr}}} e_k^+ + \mathcal{E}_{\text{ODE}}. \tag{20}$$

We want to establish a recurrence relation for e_k^+. Substitute $Eq(2)$ (with k replaced by $k+1$ for e_{k+1}^-) into $Eq(1)$ (for e_{k+1}^+):

$$e_{k+1}^+ \leq \kappa e_{k+1}^- + \delta_{\mathcal{C}} \leq \kappa (e^{L_{\mathcal{F}} \Delta t_{\text{corr}}} e_k^+ + \mathcal{E}_{\text{ODE}}) + \delta_{\mathcal{C}} = \kappa e^{L_{\mathcal{F}} \Delta t_{\text{corr}}} e_k^+ + \kappa \mathcal{E}_{\text{ODE}} + \delta_{\mathcal{C}}. \tag{21}$$

Let $\alpha_{\text{eff}} = \kappa e^{L_{\mathcal{F}} \Delta t_{\text{corr}}}$. Let $B = \kappa \mathcal{E}_{\text{ODE}} + \delta_{\mathcal{C}}$. Then the recurrence relation for the error after correction is:

$$e_{k+1}^+ \leq \alpha_{\text{eff}} e_k^+ + B. \tag{22}$$

This is a linear first-order non-homogeneous recurrence relation. Unrolling this for K steps, starting from e_0^+:

$$\begin{aligned} e_K^+ &\leq \alpha_{\text{eff}} e_{K-1}^+ + B \\ &\leq \alpha_{\text{eff}} (\alpha_{\text{eff}} e_{K-2}^+ + B) + B = \alpha_{\text{eff}}^2 e_{K-2}^+ + \alpha_{\text{eff}} B + B \\ &\cdots \\ &\leq \alpha_{\text{eff}}^K e_0^+ + B \sum_{j=0}^{K-1} \alpha_{\text{eff}}^j. \end{aligned} \tag{23}$$

Since $\alpha_{\text{eff}} < 1$, the geometric series sum is $\sum_{j=0}^{K-1} \alpha_{\text{eff}}^j = \frac{1-\alpha_{\text{eff}}^K}{1-\alpha_{\text{eff}}}$. So,

$$e_K^+ \leq \alpha_{\text{eff}}^K e_0^+ + B \frac{1-\alpha_{\text{eff}}^K}{1-\alpha_{\text{eff}}}. \tag{24}$$

This can be rewritten as:

$$e_K^+ \leq \alpha_{\text{eff}}^K e_0^+ + \frac{B}{1-\alpha_{\text{eff}}} - \frac{B\alpha_{\text{eff}}^K}{1-\alpha_{\text{eff}}} = \alpha_{\text{eff}}^K \left(e_0^+ - \frac{B}{1-\alpha_{\text{eff}}} \right) + \frac{B}{1-\alpha_{\text{eff}}}. \tag{25}$$

The constant C_2 defined in the theorem is $C_2 = \frac{\kappa \mathcal{E}_{\text{ODE}} + \delta_{\mathcal{C}}}{1-\alpha_{\text{eff}}} = \frac{B}{1-\alpha_{\text{eff}}}$. This is the asymptotic error floor for e_K^+. Substituting C_2 into $Eq(7)$, we get:

$$e_K^+ \leq \alpha_{\text{eff}}^K \left(e_0^+ - C_2 \right) + C_2. \tag{26}$$

This matches the form of Equation (15) in the theorem statement, if we interpret $\|e(t)\|$ as e_K^+ (error after K corrections) and $e(t_0)$ as e_0^+ (error after the initial, possibly hypothetical, correction, which serves as the starting point for this recurrence). The number of correction steps is $K = \lfloor (t-t_0)/\Delta t_{\text{corr}} \rfloor$.

Now for:

$$\|e(t)\| \leq C_1 \cdot \alpha_{\text{eff}}^{\lfloor (t-t_0)/\Delta t_{\text{corr}} \rfloor} + C_2. \tag{27}$$

From $e_K^+ \leq \alpha_{\text{eff}}^K (e_0^+ - C_2) + C_2$:

* If $e_0^+ - C_2 \geq 0$, then we can set $C_1 = e_0^+ - C_2$. The bound becomes $C_1 \alpha_{\text{eff}}^K + C_2$.
* If $e_0^+ - C_2 < 0$, then $e_0^+ < C_2$. Since $\alpha_{\text{eff}}^K > 0$, the term $\alpha_{\text{eff}}^K (e_0^+ - C_2)$ is negative.

Thus, $e_K^+ \leq C_2 - \alpha_{\text{eff}}^K (C_2 - e_0^+) < C_2$. In this case, setting $C_1 = 0$ ensures $e_K^+ \leq 0 \cdot \alpha_{\text{eff}}^K + C_2 = C_2$, which is true. So, $C_1 = \max(0, e_0^+ - C_2)$ correctly captures both cases and matches the definition in the theorem. The bound demonstrates that if $\alpha_{\text{eff}} < 1$, the term $\alpha_{\text{eff}}^K \to 0$ as $K \to \infty$. Consequently, the error e_K^+ converges towards the asymptotic error floor C_2.

G External Results

G.1 More Visualization Cases.

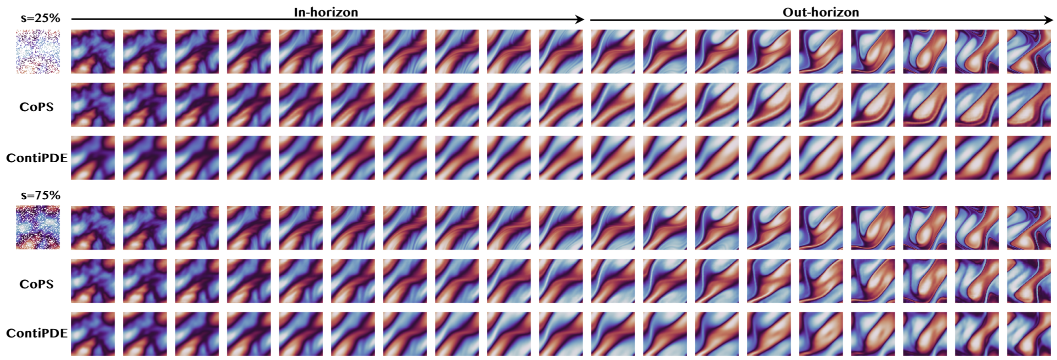

Figure 5: Qualitative comparison of our method and ContiPDE on the Navier-Stokes dataset under sparse initial observations (25% and 75%). In each panel, the first row displays the ground truth evolution, spanning both in-horizon (training) and out-horizon (extrapolation) timesteps. The second and third rows depict the predictions generated by our proposed CoPS and ContiPDE.

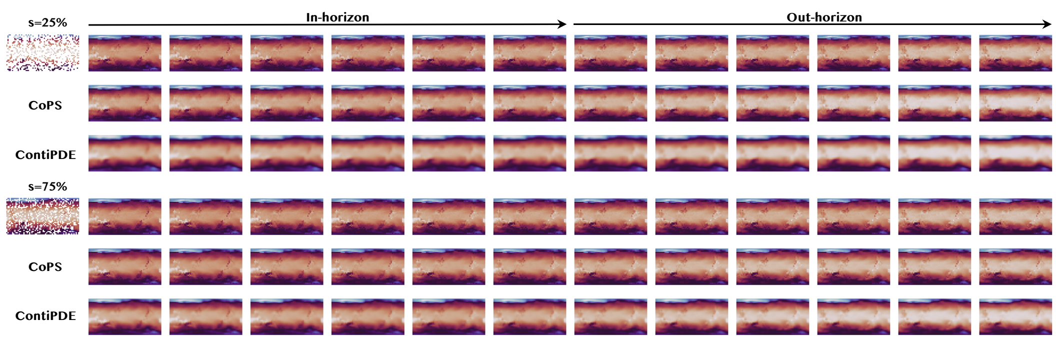

Figure 6: Qualitative comparison of our method and ContiPDE on the WeatherBench dataset under sparse initial observations (25% and 75%). In each panel, the first row displays the ground truth evolution, spanning both in-horizon (training) and out-horizon (extrapolation) timesteps. The second and third rows depict the predictions generated by our proposed CoPS and ContiPDE.

To more clearly demonstrate the performance of our method, we present detailed visualizations here. Figure 5 and Figure 6 provide qualitative illustrations of CoPS's capabilities in modeling complex fluid dynamics from sparse initial data on the Navier-Stokes dataset and the WeatherBench dataset, comparing its performance against ground truth and the ContiPDE baseline. We examine scenarios with initial conditions derived from both 25% and 75% of the full observations. The ground truth (first row in each panel) clearly shows the evolution of intricate flow patterns, including vortex formation and propagation, across both in-horizon and out-of-horizon time steps. When initialized with only 25% observations (top panel), our CoPS model (second row) successfully captures the essential dynamics, maintaining structural integrity and accurately predicting the advection of vortices well into the extrapolation phase. In contrast, ContiPDE (third row), while capturing the general flow, struggles more with the sparsity, leading to predictions that are smoother and lose some of the high-frequency details present in the ground truth, particularly in the out-of-horizon predictions. This suggests CoPS's encoding mechanism and the interplay between its continuous ODE evolution and discrete auto-correction are more effective in inferring the complete state from limited information and robustly propagating it. Even with 75% initial data (bottom panel), where both models perform better, CoPS consistently exhibits a closer match to the ground truth's finer details and long-term behavior, highlighting its enhanced capacity for accurate and stable long-range forecasting in continuous spatio-temporal domains.

G.2 Hyperparameter Analysis

We conduct sensitivity experiments with regard to correction hyperparameter (λ) on Navier-Stokes and Prometheus datasets. To resolve your concern, we conduct experiments on both In-t and Ext-t settings with observation subsampling ratio of 50%. The results on Ext-t setting demonstrate the long-term prediction performance. The results are shown in Table 4, which indicate that neural auto-correction can indeed improve the performance of our method, and the experimental results are robust to hyperparameter λ.

Table 4: Hyperparameter sensitivity of λ on the Navier-Stokes and Prometheus datasets with 50% subsampling ratio. We report MSE for In-t and Ext-t settings.

	$\lambda = 0$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$	$\lambda = 1.0$
Navier-Stokes (In-t)	3.244E-03	3.017E-03	2.925E-03	2.832E-03	2.964E-03
Navier-Stokes (Ext-t)	6.635E-03	6.172E-03	5.873E-03	5.764E-03	5.828E-03
Prometheus (In-t)	3.623E-03	3.542E-03	3.495E-03	3.374E-03	3.545E-03
Prometheus (Ext-t)	7.016E-03	6.823E-03	6.747E-03	6.678E-03	6.837E-03

G.3 Noise Disturbance Robustness.

To evaluate the robustness of our model, we present the effects of observational noise on its performance and compare these results with those of other models. We quantify the noise using the channel-specific standard deviation tailored to the dataset and have trained the model under various noise intensities (noise ratios set at 1%, 5%, 10%, 15%, and 20%). Experiments are conducted on and Navier-Stokes and Prometheus datasets. Observations from Figure 7 reveal that the proposed `CoPS` effectively maintains its performance with noise ratio below 5%, and demonstrates significant advantages over other baseline models when noise ratio increases over 10%. In contrast, we observe a more pronounced performance degradation in the two interpolation-based baseline models as noise ratio raises over 10%, which indicates their weaker robustness against noise interference.

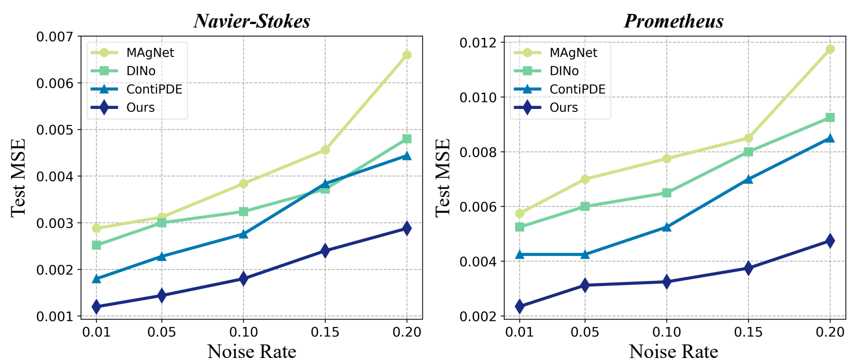

Figure 7: Performance with regard to noise disturbance.