

# Physics-aware Spatiotemporal Modules with Auxiliary Tasks for Meta-Learning

Sungyong Seo, Chuizheng Meng, Sirisha Rambhatla and Yan Liu

University of Southern California, USA

{sungyons, chuizhem, sirishar, yanliu.cs}@usc.edu

## Abstract

Modeling the dynamics of real-world physical systems is critical for spatiotemporal prediction tasks, but challenging when data is limited. The scarcity of real-world data and the difficulty in reproducing the data distribution hinder directly applying meta-learning techniques. Although the knowledge of governing partial differential equations (PDE) of data can be helpful for the fast adaptation to few observations, it is mostly infeasible to exactly find the equation for observations in real-world physical systems. In this work, we propose a framework, physics-aware meta-learning with auxiliary tasks, whose spatial modules incorporate PDE-independent knowledge and temporal modules utilize the generalized features from the spatial modules to be adapted to the limited data, respectively. The framework is inspired by a local conservation law expressed mathematically as a continuity equation and does not require the exact form of governing equation to model the spatiotemporal observations. The proposed method mitigates the need for a large number of real-world tasks for meta-learning by leveraging spatial information in simulated data to meta-initialize the spatial modules. We apply the proposed framework to both synthetic and real-world spatiotemporal prediction tasks and demonstrate its superior performance with limited observations.

## 1 Introduction

Deep learning has recently shown promise to play a major role in devising new solutions to applications with natural phenomena, such as climate change [Drgona *et al.*, 2019], ocean dynamics [Cosne *et al.*, 2019], air quality [Soh *et al.*, 2018], and so on. Deep learning techniques inherently require a large amount of data for effective representation learning, so their performance is significantly degraded when there are only a limited number of observations. However, in many tasks in the real-world physical systems we only have access to a limited amount of data. One example is air quality monitoring [Berman, 2017], in which the sensors are irregularly distributed over the space – many sensors are lo-

cated in urban areas whereas there are much fewer sensors in vast rural areas. Another example is extreme weather modeling and forecasting, i.e., temporally short events (e.g., tropical cyclones [Racah *et al.*, 2017]) without sufficient observations over time. Moreover, inevitable missing values from sensors [Tang *et al.*, 2019] further reduce the number of operating sensors and shorten the length of fully-observed sequences. Thus, achieving robust performance from a few spatiotemporal observations in physical systems remains an essential but challenging problem.

Learning on a limited amount of data from physical systems can be considered as a few shot learning. While many meta-learning techniques [Ravi and Larochelle, 2017; Finn *et al.*, 2017; Munkhdalai and Yu, 2017; Duan *et al.*, 2017; Snell *et al.*, 2017; Mishra *et al.*, 2018] have been developed to handle this few shot learning setting, there are still some challenges for the existing meta-learning methods to be applied in modeling natural phenomena. First, it is not easy to find a set of similar meta-tasks, which provide shareable latent representations needed to understand targeted observations. For instance, while image-related tasks (object detection [He *et al.*, 2017]) can take advantage of an image-feature extractor pre-trained by a large set of images [Deng *et al.*, 2009] and well-designed architecture [He *et al.*, 2016], there is no such large data corpus that is widely applicable for understanding natural phenomena. Second, unlike computer vision or natural language processing tasks where a common object (images or words) is clearly defined, it is not straightforward to find analogous objects in the spatiotemporal data. Finally, exact equations behind natural phenomena are usually unknown, causing the difficulty in reproducing the similar dataset via simulation. Although there have been some works [de Bezenac *et al.*, 2018; Greydanus *et al.*, 2019] improving data efficiency via explicitly incorporating PDEs as neural network layers when modeling spatiotemporal dynamics, it is hard to generalize for modeling different or unknown dynamics, which is ubiquitous in real-world scenario.

In this work, we propose physics-aware modules designed for meta-learning to tackle the few shot learning challenges in physical observations. One of fundamental equations in physics describing the transport of physical quantity over space and time is a continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot J = \sigma, \quad (1)$$

where  $\rho$  is the amount of the target quantity ( $u$ ) per unit volume,  $J$  is the flux of the quantity, and  $\sigma$  is a source or sink, respectively. This fundamental equation can be used to derive more specific transport equations such as the convection-diffusion equation, Navier-Stokes equations, and Boltzmann transport equation. Thus, the continuity equation is a starting point to model spatiotemporal (conservative) observations which are accessible from sensors. Based on the form of  $\rho$  and  $J$  in terms of  $u$ , Eq. 1 can be generalized as:

$$\frac{\partial u}{\partial t} = F(\nabla u, \nabla^2 u, \dots), \quad (2)$$

where the function  $F(\cdot)$  describes how the target  $u$  is changed over time from its spatial derivatives. Inspired by the form of Eq. 2, we propose two modules: spatial derivative modules (SDM) and time derivative modules (TDM). Since the spatial derivatives such as  $\nabla$ ,  $\nabla \cdot$ , and  $\nabla^2$  are commonly used across different PDEs, the spatial modules are PDE-independent and they can be meta-initialized from synthetic data. Then, the PDE-specific temporal module is trained to learn the unknown function  $F(\cdot)$  from few observations in the real-world physical systems.

This approach can effectively leverage a large amount of simulated data to train the spatial modules as the modules are PDE-independent and thus mitigating the need for a large amount of real-world tasks to extract shareable features. In addition, since the spatial modules are universally used in physics equations, the representations from the modules can be conveniently integrated with data-driven models for modeling natural phenomena. Based on the modularized PDEs, we introduce a novel approach that marries physics knowledge in spatiotemporal prediction tasks with meta-learning by providing shareable modules across spatiotemporal observations in the real-world.

Our contributions are summarized below:

- **Modularized PDEs and auxiliary tasks:** Inspired by forms of PDEs in physics, we decompose PDEs into shareable (spatial) and adaptation (temporal) parts. The shareable one is PDE-independent and specified by auxiliary tasks: *supervision of spatial derivatives*.
- **Physics-aware meta-learning:** We provide a framework for physics-aware meta-learning, which consists of PDE-independent/-specific modules. The framework is flexible to be applied to the modeling of different and unknown dynamics.
- **Utilizing synthetic data for shareable modules:** We extract shareable parameters in the spatial modules from synthetic data, which can be easily generated.

## 2 Modularized PDEs and Meta-Learning

In this section, we describe how the physics equations for conserved quantities are decomposable into two parts and how the meta-learning approach tackles the task by utilizing synthetic data when the data are limited.

### 2.1 Decomposability of Variants of a Continuity Equation

In physics, a continuity equation (Eq. 1) describes how a locally conserved quantity such as temperature, fluid den-

sity, heat, and energy is transported across space and time. This equation underlies many specific equations such as the convection-diffusion equation and Navier-Stokes equations:

$$\begin{aligned} \dot{u} &= \nabla \cdot (D\nabla u) - \nabla \cdot (\mathbf{v}u) + R, \\ \dot{\mathbf{u}} &= -(\mathbf{u} \cdot \nabla)\mathbf{u} + \nu\nabla^2\mathbf{u} - \nabla\omega + \mathbf{g}. \end{aligned}$$

where the scalar  $u$  and vector field  $\mathbf{u}$  are the variables of interest (e.g., temperature, flow velocity, etc.). A dot over a variable is time derivative. The common feature in these equations is that the forms of equations can be digested as [Bar-Sinai *et al.*, 2019]:

$$\dot{u} = F(u_x, u_y, u_{xx}, u_{yy}, \dots), \quad (3)$$

where the right-hand side denotes a function of spatial derivatives. As the time derivative can be seen as a Euler discretization [Chen *et al.*, 2018], it is notable that the next state is a function of the current state and spatial derivatives. Thus, knowing spatial derivatives at time  $t$  is a key step for spatiotemporal prediction at time  $t + 1$  for locally conserved quantities. According to Eq. 3, the spatial derivatives are universally used in variants of Eq. 1 and only the updating function  $F(\cdot)$  is specifically defined for a particular equation. This property implies that PDEs for physical quantities can be decomposable into two modules: spatial and temporal derivative modules.

### 2.2 Spatial Derivative Modules: PDE-independent Modules

Finite difference method (FDM) is widely used to discretize a  $d$ -order derivative as a linear combination of function values on a  $n$ -point stencil.

$$\frac{\partial^d u}{\partial x^d} \approx \sum_{i=1}^n \alpha_i u(x_i), \quad (4)$$

where  $n > d$ . According to FDM, it is independent for a form of PDE to compute spatial derivatives, which are input components of  $F(\cdot)$  in Eq. 3. Thus, we can modularize spatial derivatives as PDE-independent modules. The modules that can be learnable as a data-driven manner to infer the coefficients ( $\alpha_i$ ) have been proposed recently [Bar-Sinai *et al.*, 2019]. The data-driven coefficients are particularly useful when the discretization in the  $n$ -point stencil is irregular and low-resolution where the fixed coefficients cause substantial numerical errors.

### 2.3 Time Derivative Module: PDE-specific Module

Once  $d$ -order derivatives are modularized by learnable parameters, the approximated spatial derivatives from the spatial modules are fed into an additional module to learn the function  $F(\cdot)$  in Eq. 3. This module is PDE-specific as the function  $F$  describes how the spatiotemporal observations change. Since the exact form of a ground truth PDE is not given, the time derivative module is data-driven and will be adapted to observations instead.

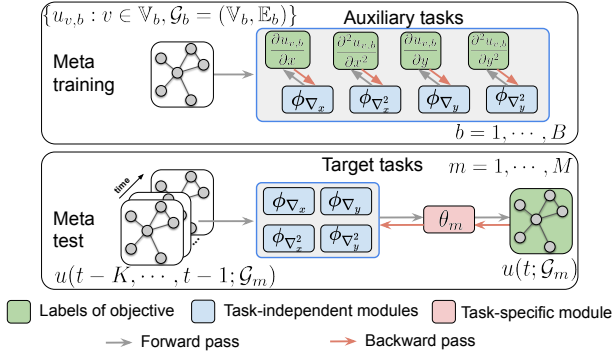


Figure 1: Overview of the physics-aware meta-learning (PiMetaL).

## 2.4 Meta-Learning with PDE-independent/-specific Modules

Recently, [Raghu *et al.*, 2020] investigate the effectiveness of model agnostic meta-learning (MAML, [Finn *et al.*, 2017]) and it is found that the outer loop of MAML is more likely to learn parameters for reusable features rather than rapid adaptation. The finding that feature reuse is the predominant reason for efficient learning of MAML allows us to use additional information which is beneficial for learning better representations. Previously, the objective in meta-training has been considered to be matched with one in meta-test as the purpose of meta-learning is to learn good initial parameters applicable across similar tasks (e.g., image classification to image classification). We are now able to incorporate auxiliary tasks under a meta-learning setting to reinforce reusable features for a main task. As described in Sec. 2.1, the spatial modules are reusable across different observations, and thus, we can meta-initialize the spatial modules first with spatial derivatives provided by synthetic datasets. Then, we can integrate the spatial modules with the task-specific temporal module during meta-test to help adaptation of TDM on few observations. Since the spatial modules are trained by readily available synthetic datasets, a large number of real-world datasets for meta-training is not required.

## 3 Physics-aware Meta-Learning with Auxiliary Tasks

In this section, we develop a physics-aware meta-learning framework for the modularized PDEs. Fig. 1 describes the proposed framework and its computational process.

### 3.1 Spatial Derivative Module

As we focus on the modeling and prediction of sensor-based observations, where the available data points are inherently on a spatially sparse irregular grid, we use graph networks for each module  $\phi_k$  to learn the finite difference coefficients. Given a graph  $\mathcal{G} = (\mathbb{V}, \mathbb{E})$  where  $\mathbb{V} = \{1, \dots, N\}$  and  $\mathbb{E} = \{(i, j) : i, j \in \mathbb{V}\}$ , a node  $i$  denotes a physical location  $\mathbf{x}_i = (x_i, y_i)$  where a function value  $u_i = u(x_i, y_i)$  is observed. Then, the graph signals with positional relative displacement as edge features are fed into the spatial modules to approximate spatial derivatives by Alg. 1. The coefficients  $(a_i, b_{(i,j)})$  on each node  $i$  and edge  $(i, j)$  are output of  $\phi$  and

### Algorithm 1 Spatial derivative module (SDM)

**Input:** Graph signals  $u_i$  and edge features  $e_{ij} = \mathbf{x}_j - \mathbf{x}_i$  on  $\mathcal{G}$  where  $\mathbf{x}_i$  is a coordinate of node  $i$ .

**Output:** Spatial derivatives  $\{\hat{u}_{k,i} \mid i \in \mathbb{V} \text{ and } k \in \mathbb{K}\}$  where  $\mathbb{K} = \{\nabla_x, \nabla_y, \nabla_x^2, \nabla_y^2\}$ .

**Require:** Spatial derivative module  $\{\phi_k \mid k \in \mathbb{K}\}$ .

- 1: **for**  $k \in \mathbb{K}$  **do**
- 2:      $\{a_{k,i}, b_{k,(i,j)} \mid (i, j) \in \mathbb{E}, k \in \mathbb{K}\} = \phi_k(\{u\}, \{e\}, \mathcal{G})$
- 3:     **for**  $i \in \mathbb{V}$  **do**
- 4:          $\hat{u}_{k,i} = a_{k,i}u_i + \sum_{(j,i) \in \mathbb{E}} b_{k,(j,i)}u_j$
- 5:     **end for**
- 6: **end for**

### Algorithm 2 Time derivative module (TDM)

**Input:** Graph signals  $u$  and approximated spatial derivatives  $\hat{u}_k$  where  $k \in \mathbb{K}$  on  $\mathcal{G}$ . Time interval  $\Delta t$

**Output:** Prediction of signals at next time step  $\hat{u}(t)$

**Require:** Time derivative module.

- 1:  $\hat{u}_t = \text{TDM}(\{u_i, \hat{u}_{k,i} \mid i \in \mathbb{V} \text{ and } k \in \mathbb{K}\})$
- 2:  $\hat{u}(t) = u(t-1) + \hat{u}_{t-1} \cdot \Delta t$

they are linearly combined with the function values  $u_i$  and  $u_j$ .  $\mathbb{K}$  denotes a set of finite difference operators. For example, if we set  $\mathbb{K} = \{\nabla_x, \nabla_y, \nabla_x^2, \nabla_y^2\}$ , we have 4 modules which approximate first/second order of spatial derivatives in 2-dimension, respectively.

### 3.2 Time Derivative Module

Once spatial derivatives are approximated, another learnable module is required to combine them for a target task. The form of line 2 in Alg. 2 comes from Eq. 3 and TDM is adapted to learn the unknown function  $F(\cdot)$  in the equation. As our target task is the regression of graph signals, we use a recurrent graph network for TDM.

### 3.3 Meta-Learning with Auxiliary Objective

As discussed in Sec. 2.1, it is important to know spatial derivatives at time  $t$  to predict next signals at  $t+1$  for locally conserved physical quantities. However, it is impractical to access the spatial derivatives in the sensor-based observations as they are highly discretized over space.

The meta-initialization with the auxiliary tasks from synthetic datasets is particularly important to address the challenge. First, the spatial modules can be universal feature extractors for modeling observations following unknown physics-based PDEs. Unlike other domains such as computer vision, it has been considered that there is no particular shareable architecture for learning spatiotemporal dynamics from physical systems. We propose that the PDE-independent spatial modules can be applicable as feature extractors across different dynamics as long as the dynamics follow a local form of conservation laws. Second, we can utilize synthetic data to meta-train the spatial modules as they are PDE-agnostic. This property allows us to utilize a large amount of synthetic datasets which are readily generated by numerical methods

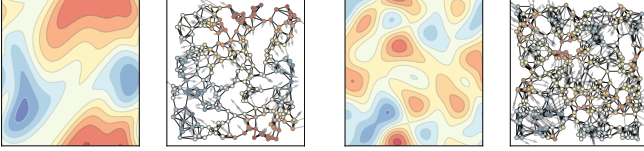


Figure 2: Examples of generated spatial function values and graph signals. Node and edge features (function value and relative displacement, respectively) are used to approximate spatial derivatives (arrows). We can adjust the number of nodes (spatial resolution), the number of edges (discretization), and the degree of fluctuation (scale of derivatives) to differentiate meta-train tasks.

---

**Algorithm 3** Meta-initialization with auxiliary tasks: Supervision of spatial derivatives

---

**Input:** A set of meta-train task datasets  $\mathcal{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_B\}$  where  $\mathcal{D}_b = (\mathcal{D}_b^{tr}, \mathcal{D}_b^{te})$ .

$\mathcal{D}_b = \{(u_i^b, e_{ij}^b, y_i^{(a_1, b)}, \dots, y_i^{(a_K, b)}) : i \in \mathbb{V}_b, (i, j) \in \mathbb{E}_b\}$  where  $y_i^{(a_k, \cdot)}$  is an  $k$ -th auxiliary task label for the  $i$ -th node.

Learning rate  $\alpha$  and  $\beta$ .

**Output:** Meta-initialized spatial modules  $\Phi$ .

- 1: Initialize auxiliary modules  $\Phi = (\phi_1, \dots, \phi_K)$
  - 2: **while** not converged **do**
  - 3:   **for**  $\mathcal{D}_b$  in  $\mathcal{D}$  **do**
  - 4:      $\Phi'_b = \Phi - \alpha \nabla_{\Phi} \sum_{k=1}^K \mathcal{L}_k^{aux}(\mathcal{D}_b^{tr}; \phi_k)$
  - 5:   **end for**
  - 6:    $\Phi \leftarrow \Phi - \beta \nabla_{\Phi} \sum_{b=1}^B \sum_{k=1}^K \mathcal{L}_k^{aux}(\mathcal{D}_b^{te}; \phi_{b,k})$
  - 7: **end while**
- 

regardless of the exact form of PDE for target observations. Finally, we can provide a stronger inductive bias which is beneficial for modeling real-world observations but not available in the observations explicitly.

Alg. 3 describes how the spatial modules are meta-initialized by MAML under the supervision of  $K$  different spatial derivatives. First, we generate values and spatial derivatives on a 2D regular grid from an analytical function. Then, we sample a finite number of points from the regular grid to represent discretized nodes and build a graph from the sampled nodes. Each graph signal and its discretization becomes input feature of a meta-train task and corresponding spatial derivatives are the auxiliary task labels. Fig. 2 visualizes graph signals and spatial derivatives for meta-initialization.

Once the spatial modules are initialized throughout meta-training, we reuse the modules for meta-test where the temporal module (the head of the network) are adapted on few observations from real-world sensors (Alg. 4). Although the standard MAML updates the body of the network (the spatial modules) as well, we only adapt the head layer ( $\theta$ ) as like almost-no-inner-loop method in [Raghu *et al.*, 2020]. The task at test time is graph signal prediction and the temporal modules ( $\theta$ ) are adapted by a regression loss function  $\mathcal{L} = \sum_{t=1}^T \|u(t) - \hat{u}(t)\|^2$  on length  $T$  sequence ( $\mathcal{D}_m^{tr}$ ) and evaluated on held-out ( $t > T$ ) sequence ( $\mathcal{D}_m^{te}$ ) with the adapted parameters.

---

**Algorithm 4** Adaptation on meta-test tasks

---

**Input:** A set of meta-test task datasets  $\mathcal{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_M\}$  where  $\mathcal{D}_m = (\mathcal{D}_m^{tr}, \mathcal{D}_m^{te})$ . Meta-initialized SDM ( $\Phi$ ).

Learning rate  $\alpha$ .

**Output:** Adapted TDM  $\theta'_m$  on the  $m$ -th task.

- 1: Initialize temporal modules ( $\theta_1, \dots, \theta_M$ )
  - 2: **for**  $\mathcal{D}_m$  in  $\mathcal{D}$  **do**
  - 3:    $\theta'_m = \theta_m - \alpha \nabla_{\theta_m} \mathcal{L}(\mathcal{D}_m^{tr}; \Phi, \theta_m)$
  - 4: **end for**
- 

	Meta-train	Meta-test
# nodes ( $N$ )	{256, 625}	{450, 800}
# edges per a node ( $E$ )	{4, 8}	{3, 6, 10}
Initial frequency ( $F$ )	{2, 5}	{3, 7}

Table 1: Parameters for synthetic dataset.

## 4 Spatial Derivative Modules: Reusable Modules

We have claimed that the spatial modules provide reusable features associated with spatial derivatives such as  $\nabla_x u$ ,  $\nabla_y u$ , and  $\nabla_x^2 u$  across different dynamics or PDEs. While it has been shown that the data-driven approximation of spatial derivatives is more precise than that of finite difference method [Seo *et al.*, 2020; Bar-Sinai *et al.*, 2019], it is not guaranteed that the modules effectively provide transferrable parameters for different spatial resolution, discretization, and fluctuation of function values. We explore whether the proposed spatial derivative modules based on graph networks can be used as a feature provider for different spatial functions and discretization.

We perform two sets of experiments: evaluate few-shot learning performance (1) when SDM is trained from scratch; (2) when SDM is meta-initialized. Fig. 2 shows how the graph signal and its discretization is changed over the different settings. If the number of nodes is large, it can provide spatially high-resolution and thus, the spatial derivatives can be more precisely approximated. Table 1 shows the parameters we used to generate synthetic datasets. Note that meta-test data is designed to evaluate interpolation/extrapolation properties. Initial frequency decides the degree of fluctuation (In Fig. 2, the right one has higher  $F$  than that of the left one.). For each parameter combination, we generate 100 different snapshots based on a form in [Long *et al.*, 2018]:

$$u_i = \sum_{|k|, |l| \leq F} \lambda_{k,l} \cos(kx_i + ly_i) + \gamma_{k,l} \sin(kx_i + ly_i), \quad (5)$$

where  $\lambda_{k,l}, \gamma_{k,l} \sim \mathcal{N}(0, 0.02)$ . The index  $i$  denotes the  $i$ -th node whose coordinate is  $(x_i, y_i)$  in the 2D space ( $[0, 2\pi] \times [0, 2\pi]$ ) and  $k, l$  are randomly sampled integers. From the synthetic data, the first- and second-order derivatives are analytically given and SDM is trained to approximate them.

The prediction results for spatial derivatives are shown in Table 2. The results show that the proposed module (SDM) is efficiently adaptable to different configuration on few samples from meta-initialized parameters compared to learning

$(N, E, F)$	(450,3,3)	(450,3,7)	(450,6,3)	(450,6,7)	(450,10,3)	(450,10,7)
SDM (from scratch)	1.337±0.044 7.278±0.225	7.111±0.148 51.544±0.148	1.152±0.043 5.997±0.083	7.206±0.180 47.527±0.768	1.112±0.036 5.353±0.193	7.529±0.241 47.356±0.560
SDM (pretrained)	<b>1.075±0.005</b> <b>6.482±0.207</b>	<b>5.528±0.010</b> <b>46.254±0.262</b>	<b>0.836±0.002</b> <b>5.251±0.245</b>	<b>5.354±0.001</b> <b>42.243±0.420</b>	<b>0.782±0.006</b> <b>4.728±0.244</b>	<b>5.550±0.012</b> <b>42.754±0.442</b>
$(N, E, F)$	(800,3,3)	(800,3,7)	(800,6,3)	(800,6,7)	(800,10,3)	(800,10,7)
SDM (from scratch)	1.022±0.030 7.196±0.159	5.699±0.242 49.602±0.715	0.789±0.021 5.386±0.136	5.179±0.069 42.509±1.080	0.718±0.010 4.536±0.204	5.517±0.110 39.642±1.173
SDM (pretrained)	<b>0.927±0.006</b> <b>6.553±0.193</b>	<b>4.415±0.011</b> <b>44.591±0.002</b>	<b>0.656±0.008</b> <b>4.960±0.266</b>	<b>3.977±0.025</b> <b>37.629±0.760</b>	<b>0.570±0.006</b> <b>4.213±0.275</b>	<b>4.107±0.019</b> <b>35.849±0.947</b>

Table 2: Prediction error (MAE) of the first (top) and second (bottom) order spatial derivatives.

from scratch. The finding implies that the parameters for spatial derivatives can be generally applicable across different spatial resolution, discretization, and function fluctuation.

## 5 Experimental Evaluation

### 5.1 Preliminary: Which Synthetic Dynamics Need to Be Generated?

While Table 2 demonstrates that the PDE-independent representations are reusable across different configurations, it is still an open question: which topological configuration needs to be used to construct the synthetic dynamics? According to Table 2, the most important factor affecting error is an initial frequency ( $F$ ), which determines min/max scales and fluctuation of function values, and it implies that the synthetic dynamics should be similarly scaled to a target dynamics. We use the same topological configuration in Table 1 to generate synthetic datasets for a task in Sec. 5.2 and Sec. 5.3.

### 5.2 Multi-step Graph Signal Generation

**Task.** We adopt a set of multi-step spatiotemporal sequence generation tasks to evaluate our proposed framework. In each task, the data is a sequence of  $L$  frames, where each frame is a set of observations on  $N$  nodes in space. Then, we train an auto-regressive model with the first  $T$  frames ( $T$ -shot) and generate the following  $L - T$  frames repeatedly from a given initial input ( $T$ -th frame) to evaluate its performance.

**Datasets.** For all experiments, we generate meta-train tasks with the parameters described in Table 1 and the target observations are 2 real-world datasets: (1) **AQI-CO**: national air quality index (AQI) observations [Berman, 2017]; (2) **ExtremeWeather (EW)**: the extreme weather dataset [Racah *et al.*, 2017]. For the AQI-CO dataset, we construct 12 meta-test tasks with the carbon monoxide (CO) ppm records from the first week of each month in 2015 at land-based stations. For the extreme weather dataset, we select the top-10 extreme weather events with the longest lasting time from the year 1984 and construct a meta-test task from each event with the observed surface temperatures at randomly sampled locations. Since each event lasts fewer than 20 frames, each task has a very limited amount of available data. In both datasets, graph signals are univariate. Note that all quantities have fluidic properties such as diffusive and convection. More details are in the supplementary material.

$T$ -shot	Method	AQI-CO	EW
5-shot	FDM+RGN (scratch)	0.029±0.004	0.988±0.557
	PA-DGN (scratch)	0.036±0.009	0.965±0.138
	PiMetaL (meta-init)	<b>0.025±0.006</b>	<b>0.917±0.075</b>
7-shot	FDM+RGN (scratch)	0.026±0.002	0.763±0.060
	PA-DGN (scratch)	0.023±0.002	0.748±0.020
	PiMetaL (meta-init)	<b>0.018±0.002</b>	<b>0.727±0.009</b>
10-shot	FDM+RGN (scratch)	0.021±0.001	0.709±0.003
	PA-DGN (scratch)	0.015±0.001	0.416±0.015
	PiMetaL (meta-init)	<b>0.012±0.001</b>	<b>0.407±0.025</b>

Table 3: Multi-step prediction results (MSE) and standard deviations on the two real-world datasets.

**Baselines.** We evaluate the performance of a physics-aware architecture (**PA-DGN**) [Seo *et al.*, 2020], which also consists of spatial derivative modules and recurrent graph networks (RGN), to see how the additional spatial information affects prediction performance for same architecture. Note that PA-DGN has same modules in PiMetaL and the difference is that PiMetaL utilizes meta-initialized spatial modules and PA-DGN is randomly initialized for learning from scratch on meta-test tasks. Additionally, the spatial modules in PA-DGN is replaced by finite difference method (**FDM+RGN**) to see if the numerical method provides better PDE-agnostic representations. The baselines and PiMetaL are trained on the meta-test support set only to demonstrate how the additional spatial information is beneficial for few-shot learning tasks.

**Discussion.** Table 3 shows the multi-step prediction performance of our proposed framework against the baselines on real-world datasets. Overall, PA-DGN and PiMetaL show similar trend such that the prediction error is decreased as longer series are available for few-shot adaptation. There are two important findings: first, with the similar expressive power in terms of the number of learnable parameters, the meta-initialized spatial modules provide high quality representations which are easily adaptable across different spatiotemporal dynamics in the real-world. This performance gap demonstrates that we can get a stronger inductive bias from synthetic datasets without knowing PDE-specific information. Second, the contribution of the meta-initialization is more significant when the length of available sequence is shorter ( $T = 5$ ) and this demonstrates when the meta-initialization is particularly effective. Finally, the finite dif-

$T$ -shot (Region)	GCN	GAT	GraphSAGE	GN	PA-DGN	PiMetaL
5-shot (USA)	2.742±0.120	2.549±0.115	2.128±0.146	2.252±0.131	1.950±0.152	<b>1.794±0.130</b>
10-shot (USA)	2.371±0.095	2.178±0.066	1.848±0.206	1.949±0.115	1.687±0.104	<b>1.567±0.103</b>
5-shot (EU)	1.218±0.218	1.161±0.234	1.165±0.248	1.181±0.210	0.914±0.167	<b>0.781±0.019</b>
10-shot (EU)	1.186±0.076	1.142±0.070	1.044±0.210	1.116±0.147	0.831±0.058	<b>0.773±0.014</b>

Table 4: Graph signal regression results (MSE,  $10^{-3}$ ) and standard deviations on the two regions of weather stations.

ference method provides proxies of exact spatial derivatives and the representations are useful particularly when  $T = 5$  but its performance is rapidly saturated and it comes from the gap between the learnable spatial modules and fixed numerical coefficients. The results provide a new point of view on how to utilize synthetic or simulated datasets to handle challenges caused by limited datasets.

### 5.3 Graph Signal Regression

**Task, datasets, and baselines.** [Defferrard *et al.*, 2019] conducted a graph signal regression task: predict the temperature  $x_t$  from the temperature on the previous 5 days ( $x_{t-5} : x_{t-1}$ ). We split the **GHCN** dataset spatially into two regions: (1) the USA (1,705 stations) and (2) Europe (EU) (703 stations) where there are many weather stations full functioning. In this task, the number of shots is defined as the number of input and output pairs to train a model.

As the input length is fixed, more variants of graph neural networks are considered as baselines. We concatenate the 5-step signals and feed it into Graph convolutional networks (**GCN**) [Kipf and Welling, 2017], Graph attention networks (**GAT**) [Veličković *et al.*, 2018], **GraphSAGE** [Hamilton *et al.*, 2017], and Graph networks (**GN**) [Sanchez-Gonzalez *et al.*, 2018] to predict next signals across all nodes.

**Discussion.** Table 4 shows the results of the graph signal regression task across different baselines and the proposed method. There are two patterns in the results. First, although in general we observe an improvement in performance for all methods when we move from the 5-shot setting to the 10-shot setting, PiMetaL’s performance yields the smallest error. Second, for the EU dataset, while 5-shot seems enough to achieve stable performance, it demonstrates that the PDE-independent representations make the regression error converge to a lower level. Overall, the experimental results prove that the learned spatial representations from simulated dynamics are beneficial for learning on limited data.

## 6 Related Work

**Physics-informed learning.** Since physics-informed neural networks are introduced in [Raissi *et al.*, 2019], which find that a solution of a PDE can be discovered by neural networks, physical knowledge has been used as an inductive bias for deep neural networks. Advection-diffusion equation is incorporated with deep neural networks for sea-surface temperature dynamics [de Bezenac *et al.*, 2018]. [Lutter *et al.*, 2019; Greydanus *et al.*, 2019] show that Lagrangian/Hamiltonian mechanics can be imposed to learn the equations of motion of a mechanical system and [Seo and Liu, 2019] regularize a graph neural network with a specific physics equation. Rather

than using explicitly given equations, physics-inspired inductive bias is also used for reasoning dynamics of discrete objects [Battaglia *et al.*, 2016] and continuous quantities [Seo *et al.*, 2020]. [Long *et al.*, 2018] propose a numeric-symbolic hybrid deep neural network designed to discover PDEs from observed dynamic data. To the best of our knowledge, we are the first to provide a framework to use the physics-inspired inductive bias under the meta-learning settings to tackle the limited data issue which is common for real-world data such as extreme weather events [Racah *et al.*, 2017].

**Meta-learning.** The aim of meta-learning is to enable learning parameters which can be used for new tasks unknown at the time of learning, leading to agile models which adapt to a new task utilizing only a few samples. Based on how the knowledge from the related tasks is used, meta-learning methods have been classified as optimization-based [Ravi and Larochelle, 2017; Finn *et al.*, 2017], model-based [Munkhdalai and Yu, 2017; Duan *et al.*, 2017; Mishra *et al.*, 2018], and metric-based [Snell *et al.*, 2017]. Recently, another branch of meta-learning has been introduced to more focus on finding a set of reusable modules as components of a solution to a new task. [Alet *et al.*, 2019] provide a framework, structured modular meta-learning, where a finite number of modules are introduced as task-independent modules and an optimal structure combining the modules is found from a limited number of data. To our knowledge, none of the above works provide a solution for modeling physics-related spatiotemporal dynamics where it is hard to generate enough tasks for meta-initialization.

## 7 Conclusion

In this paper, we propose a framework for physics-aware meta-learning with auxiliary tasks. By incorporating PDE-independent/ $T$ -invariant knowledge from simulated data, the framework provide reusable features to meta-test tasks with a limited amount of data. Experiments show that auxiliary tasks and physics-aware meta-learning help construct reusable modules that improve the performance of spatiotemporal predictions when data is limited.

## Acknowledgements

This work is supported in part by NSF Research Grant IIS-1254206 and MINERVA grant N00014-17-1-2281, granted to co-author Yan Liu in her academic role at the University of Southern California. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agency, or the U.S. Government.

## References

- [Alet *et al.*, 2019] Ferran Alet, Erica Weng, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Neural relational inference with fast modular meta-learning. In *Advances in Neural Information Processing Systems*, pages 11804–11815, 2019.
- [Bar-Sinai *et al.*, 2019] Yohai Bar-Sinai, Stephan Hoyer, Jason Hickey, and Michael P Brenner. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31):15344–15349, 2019.
- [Battaglia *et al.*, 2016] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Advances in neural information processing systems*, pages 4502–4510, 2016.
- [Berman, 2017] Lex Berman. National aqi observations (2014-05 to 2016-12). Harvard Dataverse, 2017.
- [Chen *et al.*, 2018] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.
- [Cosne *et al.*, 2019] Gautier Cosne, Guillaume Maze, and Pierre Tandeo. Coupling oceanic observation systems to study mesoscale ocean dynamics. *arXiv:1910.08573*, 2019.
- [de Bezenac *et al.*, 2018] Emmanuel de Bezenac, Arthur Pajot, and Patrick Gallinari. Deep learning for physical processes: Incorporating prior scientific knowledge. In *ICLR*, 2018.
- [Defferrard *et al.*, 2019] Michaël Defferrard, Martino Milani, Frédéric Gusset, and Nathanaël Perraudin. DeepSphere: a graph-based spherical cnn. In *ICLR*, 2019.
- [Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [Drgona *et al.*, 2019] Ján Drgona, Lieve Helsen, and Draguna Vrabie. Stripping off the implementation complexity of physics-based model predictive control for buildings via deep learning. *Advances in Neural Information Processing Systems*, 2019.
- [Duan *et al.*, 2017] Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. In *Advances in neural information processing systems*, 2017.
- [Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.
- [Greydanus *et al.*, 2019] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, 2019.
- [Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, 2017.
- [He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [He *et al.*, 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2017.
- [Kipf and Welling, 2017] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
- [Long *et al.*, 2018] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data. In *International Conference on Machine Learning*. PMLR, 2018.
- [Lutter *et al.*, 2019] Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. In *ICLR*, 2019.
- [Mishra *et al.*, 2018] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *ICLR*, 2018.
- [Munkhdalai and Yu, 2017] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.
- [Racah *et al.*, 2017] Evan Racah, Christopher Beckham, Tegan Maharaj, Samira Ebrahimi Kahou, Mr Prabhat, and Chris Pal. Extremeweather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. In *Advances in Neural Information Processing Systems*, pages 3402–3413, 2017.
- [Raghu *et al.*, 2020] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In *ICLR*, 2020.
- [Raissi *et al.*, 2019] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [Ravi and Larochelle, 2017] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.
- [Sanchez-Gonzalez *et al.*, 2018] Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. *arXiv:1806.01242*, 2018.
- [Seo and Liu, 2019] Sungyong Seo and Yan Liu. Differentiable physics-informed graph networks. *arXiv:1902.02950*, 2019.
- [Seo *et al.*, 2020] Sungyong Seo, Chuizheng Meng, and Yan Liu. Physics-aware difference graph networks for sparsely-observed dynamics. In *ICLR*, 2020.
- [Snell *et al.*, 2017] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.
- [Soh *et al.*, 2018] Ping-Wei Soh, Jia-Wei Chang, and Jen-Wei Huang. Adaptive deep learning-based air quality prediction model using the most relevant spatial-temporal relations. *Ieee Access*, 6:38186–38199, 2018.
- [Tang *et al.*, 2019] Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang. Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values. *arXiv:1911.10273*, 2019.
- [Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.