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

Sungyong Seo, Chuizheng Meng, Sirisha Rambhatla, Yan Liu

Department of Computer Science, University of Southern California

Motivation

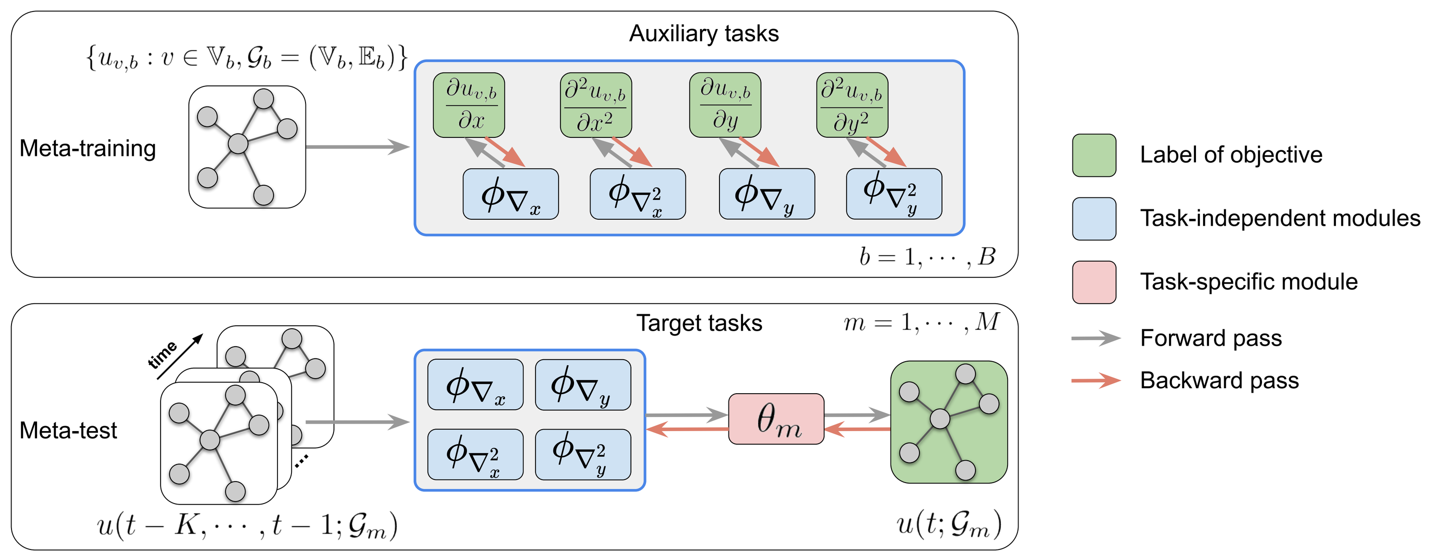
Limited Real-world Observations

- Modeling natural phenomena with deep neural networks when only a limited number of observations are available is challenging.
- The sparsely available sensor-based data cause substantial numerical error when we utilize existing differential methods.
- Temporally short events and inevitable missing values from sensors further shorten the length of fully-observed sequences.

Challenges in existing few-shot learning methods

- It is not easy to find a set of similar meta-tasks which provide shareable latent representations needed to understand targeted observations.
- Unlike computer vision or NLP tasks where a common object (images or words) is clearly defined, it is not straightforward to find analogous objects in the spatiotemporal data.
- Exact equations behind natural phenomena are usually unknown, leading to the difficulty in reproducing the similar dataset via simulation.

Proposed Architecture



Schematic overview of the physics-aware meta-learning (PiMetaL).

Contributions

Modularized PDEs and auxiliary tasks:

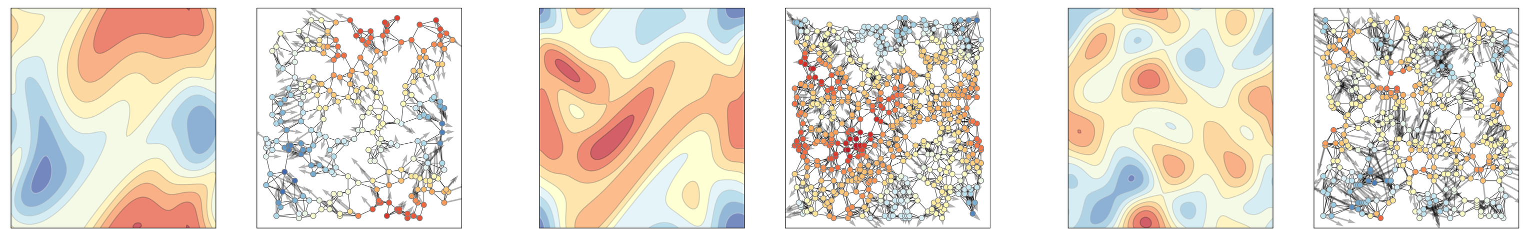
Inspired by forms of PDEs in physics, we decompose PDEs into shareable (spatial) and adaptation (temporal) parts. The shareable one is PDE-independent and specified by auxiliary tasks: *supervision of spatial derivatives*.

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 or unknown dynamics.

Synthetic data for shareable modules:

We extract shareable parameters in the spatial modules from synthetic data, which can be generated from different dynamics easily.



(a) (b) (c)

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.

Modularized PDEs

Decomposability of Variants of a Continuity Equation

- One of fundamental equations in physics describing the transport of physical quantity over space and time is a continuity equation:

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

where ρ is the amount of the target quantity (u) per unit volume, J is the flux of the quantity, and σ is a source or sink, respectively.

- Based on the form of ρ and J with respect to a particular quantity u, Eq. 1 can be generalized as:

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

where the function $F(\cdot)$ describes how the target u is changed over time from its spatial derivatives. This equation underlies many specific equations such as the convection-diffusion equation and Navier-Stokes equations:

$$\dot{u} = \nabla \cdot (D \nabla u) - \nabla \cdot (\boldsymbol{v} u) + R, \qquad \text{(Convection-Diffusion eqn.)}$$

$$\dot{\boldsymbol{u}} = -(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nu \nabla^2 \boldsymbol{u} - \nabla \omega + \boldsymbol{g}. \qquad \text{(Incompressible Navier-Stokes eqn.)}$$

- Inspired by the form of Eq. 2, we propose two modules: spatial derivative modules (SDM) and time derivative modules (TDM).

Spatial Derivative Modules (SDM): PDE-independent Modules

Finite difference method (FDM) is used to compute a d-order derivative as a linear combination of n-point stencil values.

$$\frac{\partial^d u}{\partial x^d} \approx \sum_{i=1}^{n} \alpha_i u(x_i), \tag{3}$$

where $n > d$. Since Eq. 3 is independent for a form of $F(\cdot)$ in Eq. 2, we can modularize spatial derivatives as PDE-independent modules and use them as input of $F(\cdot)$.

Time Derivative Module (TDM): PDE-specific Module

- Once upto d-order derivatives are modularized by SDM, the approximated spatial derivatives are fed into an additional module to learn the function $F(\cdot)$ in Eq. 2.
- This module is PDE-specific as the function F describes how the spatiotemporal observations change.

Spatial Derivative Modules: Reusable Modules

- We have claimed that SDM 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.
- We explore if the proposed SDM based on graph networks can be used as a feature provider for different spatial functions and discretization.

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

(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)	**1.075±0.005** **6.482±0.207**	**5.528±0.010** **46.254±0.262**	**0.836±0.002** **5.251±0.245**	**5.354±0.001** **42.243±0.420**	**0.782±0.006** **4.728±0.244**	**5.550±0.012** **42.754±0.442**
(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)	**0.927±0.006** **6.553±0.193**	**4.415±0.011** **44.591±0.002**	**0.656±0.008** **4.960±0.266**	**3.977±0.025** **37.629±0.760**	**0.570±0.006** **4.213±0.275**	**4.107±0.019** **35.849±0.947**

Experimental Results

Graph Signal Generation We adopt a set of multi-step spatiotemporal sequence generation tasks to evaluate our proposed framework on two real-world dataset (AQI-CO [2]: air quality index, ExtremeWeather [3]: the extreme weather dataset).

Table: Multi-step prediction results (MSE).

T-shot	Method	AQI-CO	ExtremeWeather
5-shot	FDM+RGN (scratch)	0.0291±0.0039	0.9883±0.5567
	PA-DGN (scratch)	0.0363±0.0090	0.9653±0.1384
	PiMetaL (meta-init)	**0.0253±0.0055**	**0.9167±0.0746**
7-shot	FDM+RGN (scratch)	0.0258±0.0023	0.7626±0.0602
	PA-DGN (scratch)	0.0225±0.0018	0.7478±0.0199
	PiMetaL (meta-init)	**0.0182±0.0019**	**0.7274±0.0089**
10-shot	FDM+RGN (scratch)	0.0213±0.0013	0.7090±0.0030
	PA-DGN (scratch)	0.0146±0.0005	0.4156±0.0145
	PiMetaL (meta-init)	**0.0115±0.0004**	**0.4066±0.0247**

Graph Signal Regression [4] 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 (Global Historical Climatology Network (GHCN) provided by National Oceanic and Atmospheric Administration (NOAA) spatially into two regions: (1) the USA (1,705 stations) and (2) Europe (EU) (703 stations) where there are many weather stations full functioning.

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

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	**1.794±0.130**
10-shot (USA)	2.371±0.095	2.178±0.066	1.848±0.206	1.949±0.115	1.687±0.104	**1.567±0.103**
5-shot (EU)	1.218±0.218	1.161±0.234	1.165±0.248	1.181±0.210	0.914±0.167	**0.781±0.019**
10-shot (EU)	1.186±0.076	1.142±0.070	1.044±0.210	1.116±0.147	0.831±0.058	**0.773±0.014**

Conclusion

In this paper, we propose a framework for physics-aware meta-learning with auxiliary tasks. By incorporating PDE-independent knowledge (spatial derivatives) from simulated data, the framework provide reusable features and the features help improve the meta-test tasks with a limited amount of data.

References

[1] Bar-Sinai et al. Learning data-driven discretizations for partial differential equations. PNAS, 2019.

[2] Berman. National aqi observations (2014-05 to 2016-12). Harvard Dataverse, 2017.

[3] Racah et al. Extremeweather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. NeurIPS 2017.

[4] Defferrard et al. DeepSphere: a graph-based spherical cnn. ICLR, 2019.