METAPHYSICA: IMPROVING OOD ROBUSTNESS IN PHYSICS-INFORMED MACHINE LEARNING

S Chandra Mouli
Department of Computer Science
Purdue University, Indiana, US

Muhammad Ashraful Alam
Department of Electrical and Computer Engineering,
Purdue University, Indiana, US

Bruno Ribeiro
Department of Computer Science
Purdue University, Indiana, US

ABSTRACT

A fundamental challenge in physics-informed machine learning (PIML) is the design of robust PIML methods for out-of-distribution (OOD) forecasting tasks. These OOD tasks require learning-to-learn from observations of the same (ODE) dynamical system with different unknown ODE parameters, and demand accurate forecasts even under out-of-support initial conditions and out-of-support ODE parameters. In this work we propose to improve the OOD robustness of PIML via a meta-learning procedure for causal structure discovery. Using three different OOD tasks, we empirically observe that the proposed approach significantly outperforms existing state-of-the-art PIML and deep learning methods (with $2\times$ to $28\times$ lower OOD errors).

1 INTRODUCTION

Physics-informed machine learning (PIML) (e.g., (Willard et al., 2020; Xingjian et al., 2015; Lusch et al., 2018; Yeo & Melnyk, 2019; Raissi et al., 2018; Kochkov et al., 2021)) seeks to combine the strengths of physics and machine learning models and has positively impacted fields as diverse as biological sciences (Yazdani et al., 2020), climate science (Faghmous & Kumar, 2014), turbulence modeling (Ling et al., 2016; Wang et al., 2020a), among others. PIML achieves substantial success in tasks where the test data comes from the same distribution as the training data (*in-distribution tasks*).

This work expands on existing PIML approaches to forecast a parametric dynamical system (ODE), with a focus on out-of-distribution (OOD) scenarios. In our tasks, OOD robustness is tied to interventions over the initial system state and the unknown ODE parameters (illustrated in Figure 1(a,b)), not arbitrary interventions as the system evolves from the initial state (see (Rubenstein et al., 2016) for the effect of arbitrary interventions in physics models). In this setting, we observe that existing state-of-the-art PIML models perform significantly worse OOD than in-distribution, with some scenarios being challenging even for PIML methods designed with OOD robustness in mind (Wang et al., 2021b; Kirchmeyer et al., 2022). This is because the standard ML part of PIML still learns spurious associations and performs poorly in our OOD setting. While there are PIML methods (Brunton et al., 2016; Martius & Lampert, 2016) robust to OOD initial conditions, they are not robust to shifts in ODE parameters as they are *transductive*, i.e., they do not combine knowledge from diverse training trajectories with different ODE parameters.

We then propose an approach for more robust dynamical system forecasting that leverages *causal structure discovery* (e.g., Zheng et al. (2018)) combined with invariant risk minimization (Arjovsky et al., 2019; Krueger et al., 2021) to learn the underlying ODE structure, and uses *meta-learning* to learn from different trajectories of the same dynamical system. More precisely, our contributions are:

1. We show that state-of-the-art PIML and deep learning methods fail in test examples with OOD initial conditions and/or OOD system parameters. Prior work (Wang et al., 2021a) showed that deep learning-only methods fail in OOD tasks, and argued physics models and PIML methods

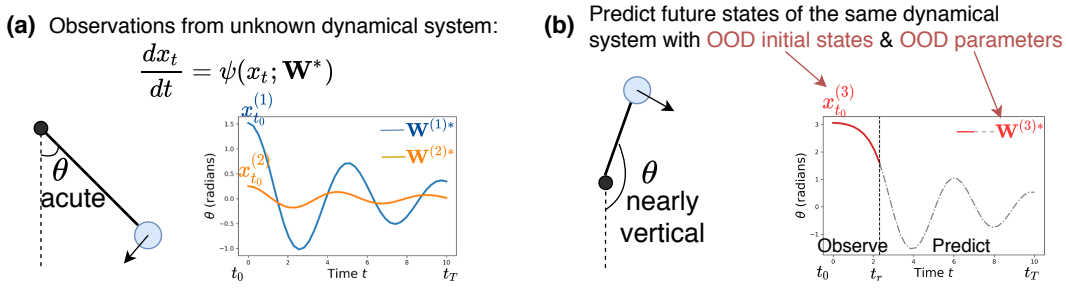


Figure 1: Dynamical system OOD problem definition. **(a)** Training data consists of multiple observations from the same dynamical system with different initial conditions and parameters. **(b)** At test, we are given observations till t_r (red solid) and the goal is to predict the future observations till t_T (gray dashed). The initial conditions and the ODE parameters can be OOD in test.

would succeed, including a proposed solution (Wang et al., 2021b). Here we show that PIML methods also fail (or perform poorly) OOD, including the solution in Wang et al. (2021b).

2. *Meta-learning framework for learning from diverse training trajectories*: By meta learning we mean the definition in (Thrun & Pratt, 1998, Chapter 1.2) adapted to our setting. Given **(a)** a family of M tasks or experiments observing the same dynamical system with different parameters, **(b)** training observations or trajectory $\mathbf{X}_{t_0}^{(i)}, \ldots, \mathbf{X}_{t_T}^{(i)}$, for each task i, and **(c)** a performance measure (prediction error) for each task; our algorithm will *meta learn* such that performance on each task improves with more observations per task and with the number of tasks. Our architecture uses parameters shared across all tasks to allow transfer of knowledge between multiple tasks.
3. *Learning the ODE via causal structure discovery*: We define a family of structural causal models, each identifying a different ODE, and perform a search to find the underlying dynamical system (assumed to be in the family). We propose a causal structure discovery approach via continuous optimization with ℓ_1-regularization and an invariant risk minimization-type penalty (Krueger et al., 2021), both of which are empirically shown to be necessary. We further show in Theorem 1 that the proposed approach learns the true causal structure under certain identifiability conditions.

The proposed method is then empirically validated using three commonly-used simulated physics tasks (with measurement noise): Damped pendulum systems (Yin et al., 2021), predator-prey systems (Wang et al., 2021a), and epidemic modeling (Wang et al., 2021a).

2 Dynamical System Forecasting: OOD Setting

We formally describe the dynamical system forecasting problem under out-of-distribution scenarios.

Definition 1 (Dynamical system forecasting problem)**.** *The dynamical system is described as an ordinary differential equation (ODE) as follows:*

$$\frac{d\boldsymbol{x}_t}{dt} = \psi(\boldsymbol{x}_t; \boldsymbol{W}^*)\,, \tag{1}$$

where $\boldsymbol{x}_t \in \mathbb{R}^d$ is the hidden state of the system at time t, ψ is an unknown deterministic function with unknown ODE parameters $\boldsymbol{W}^*$. We define noisy observations of the dynamical system $\mathbf{X}_t \sim \mathcal{N}(\boldsymbol{x}_t, \sigma_\varepsilon^2 I)$ where σ_ε^2 is the noise variance. $\boldsymbol{x}_{t_0}$ denotes the initial state at time t_0.

1. **Training data (Figure 1a):** *In training, we are given a set of M tasks (or trajectories) from the dynamical system in Equation* (1)*, $\mathcal{T}^{(i)} := \mathbf{X}_{t_0}^{(i)}, \ldots, \mathbf{X}_{t_{T^{(i)}}}^{(i)}$, $i \in \{1, \ldots, M\}$, of lengths $T^{(i)}+1$ respectively, with*
$$\boldsymbol{x}_{t_0}^{(i)} \sim P^{tr}(\boldsymbol{x}_{t_0})\,, \quad \mathbf{W}^{(i)*} \sim P^{tr}(\mathbf{W}^*)\,, \tag{2}$$
where for each task i, $\boldsymbol{x}_{t_0}^{(i)}$ is the initial state at time t_0, $\mathbf{W}^{(i)*}$ are the (hidden) task-specific ODE parameters, and $\{t_0, \ldots, t_{T^{(i)}}\}$ are regularly-spaced discrete time steps.
2. **OOD test data (Figure 1b):** *At test, we are given $(r+1)$-length initial observations $\mathcal{T}^{(te)} := \mathbf{X}_{t_0}^{(te)}, \ldots, \mathbf{X}_{t_r}^{(te)}$, from the same dynamical system in Equation* (1) *with an arbitrary initial state $\boldsymbol{x}_{t_0}^{(te)}$ and ODE parameters $\mathbf{W}^{(te)*}$ possibly outside their respective training supports, $supp(P^{tr}(\boldsymbol{x}_{t_0}))$ and $supp(P^{tr}(\mathbf{W}^*))$.*

Forecasting goal *is to predict the future observations $\mathbf{X}_{t_{r+1}}^{(te)}, \ldots, \mathbf{X}_{t_{T^{(te)}}}^{(te)}$ till some time $T^{(te)}$.*

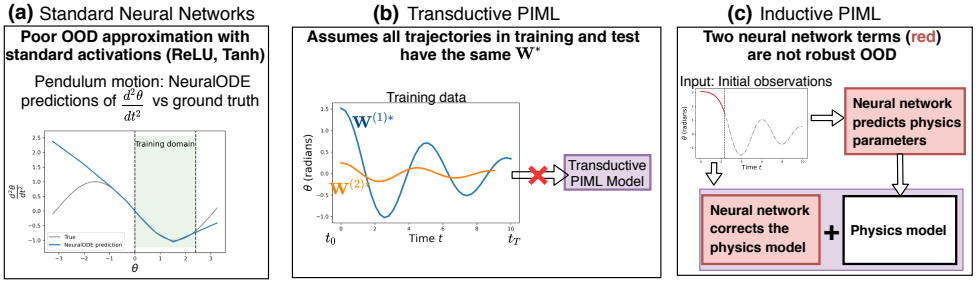


Figure 2: Shows OOD challenges of existing methods. **(a) Standard Neural networks** (e.g., (Chen et al., 2018)), when trained on the motion of damped pendulum, predict accurately in the training domain (green shaded), but predict a linear function outside the training domain. **(b) Transductive PIML methods** (e.g., (Raissi et al., 2017a; Brunton et al., 2016)) assume all training/test trajectories have the same dynamical system parameters $\boldsymbol{W}^{(i)*}$, which is not the general setting. **(c) Inductive PIML methods** (e.g., (Yin et al., 2021; Mehta et al., 2021)) use a neural network to predict parameters of a known physics model and another neural network as residual correction. These two neural networks face the same robustness issues discussed in **(a)**.

In summary, we are given training trajectories from that may have (a) different initial conditions, and (b) different unknown ODE parameters. We observe a test trajectory from time $t = t_0, \ldots, t_r$ and we wish to forecast its future after time t_r. The test trajectory can have an OOD initial condition and OOD ODE parameters.

Illustrative example. Figure 1 shows an example of an out-of-distribution task for forecasting the motion of a pendulum with friction. The state $\boldsymbol{x}_t = [\theta_t, \omega_t] \in \mathbb{R}^2$ describes the angle made by the pendulum with the vertical and the corresponding angular velocity at time t. The true (unknown) function ψ describing this dynamical system is given by $\psi([\theta_t, \omega_t]; \boldsymbol{W}^*) = [\omega_t, -\alpha^{*2}\sin(\theta_t) - \rho^*\omega_t]$ with $\boldsymbol{W}^* = (\alpha^*, \rho^*)$ denoting the ODE parameters relating to the pendulum's period and the damping coefficient. They are task-dependent, e.g., different tasks can have different damping coefficients.

1. In training, we observe M noisy trajectories (tasks) of motion over time $t = 0, 0.1, \ldots, 10$ from experiments where a pendulum is dropped from different acute angles $0 < \theta_{t_0}^{(i)} < \pi/2$ in materials with different damping coefficients $\rho^{(i)*}$.
2. In test, the pendulum is dropped from nearly vertical angles, $\pi - 0.1 < \theta_{t_0}^{(\text{te})} < \pi$ in a material with much higher damping coefficient $\rho^{(\text{te})*}$. The test trajectory is observed over a smaller time window $t \leq 3$ and the goal is to predict future pendulum states till time $t = 10$.

An ML or PIML model for this forecasting task must be able to learn from diverse trajectories of the same dynamical system, and be robust to shifts in distribution between training and test.

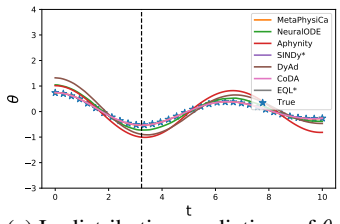
3 Related Work & Their Limitations

Next we describe different classes of existing approaches that are commonly used for the dynamical system forecasting and their inherent challenges out-of-distribution.

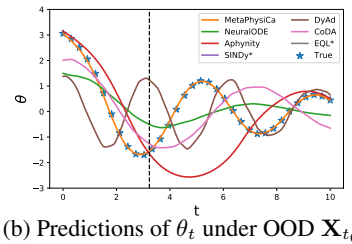
3.1 Neural Network Methods

Deep learning's ability to model complex phenomena has allowed it to make great strides in many physics applications (Lusch et al., 2018; Yeo & Melnyk, 2019; Kochkov et al., 2021; Dang et al., 2022; Brandstetter et al., 2022b). However, standard deep learning methods are known to learn spurious correlations and tend to fail when the test distribution of the inputs are different from that in training (Wang et al., 2021a; Geirhos et al., 2020). Figure 3b depicts the out-of-distribution failure of several deep learning methods from NeuralODE (Chen et al., 2018) to more complex meta learning approaches (Wang et al., 2021b) in our damped pendulum example (more details of the experiment is in Section 5). While DyAd (Wang et al., 2021b) and CoDA (Kirchmeyer et al., 2022) use meta-learning to adapt to new ODE parameters, they are not robust to OOD initial conditions.

In standard deep learning tasks, Xu et al. (2021) show that an MLP's failure to extrapolate to out-of-distribution can be traced to an absence of algorithmic alignment, which is an appropriate combination of basis and activation functions within the architecture for the task. For example, the outputs of an MLP with ReLU activations will be linear far from the training domain even when trained to predict a sine/quadratic function. For dynamical system forecasting, our Figure 2a depicts the results of a similar experiment for a standard sequence model (NeuralODE): the model can approximate the



(a) In-distribution predictions of θ_t



(b) Predictions of θ_t under OOD $\mathbf{X}_{t_0}$

Figure 3: **(Damped pendulum.)** Predict pendulum motion **(a)** in-distribution, when dropped from acute angles and **(b)** OOD w.r.t initial conditions and parameters, when a different pendulum is dropped from nearly vertical angles. Figures show ground truth curve (blue stars) in- and out-of-distribution along with predictions from different models. **Existing methods perform well in-distribution, but perform poorly OOD even in this simple example.**

target sine function in the training domain (green region) but predicts a linear function far outside the training domain. This means that **we need algorithmic alignment (i.e., to include appropriate basis functions) in order to make accurate forecasts in OOD tasks.** In Appendix D.6 we show that even our proposed method without algorithmic aligment cannot properly extrapolate OOD (but it still does better than baselines).

3.2 Physics-informed machine learning (PIML)

To alleviate the challenges of standard neural networks, several PIML methods have been proposed (e.g., (Willard et al., 2020; Wang et al., 2020a; Faghmous & Kumar, 2014; Daw et al., 2017)) that utilize physics-based domain knowledge for better predictions. The physics-based knowledge vary across methods, for example, **(a)** a dictionary of basis functions (e.g., $\sin$, $\cos$, $\frac{d}{dt}$) (Schmidt & Lipson, 2009; Brunton et al., 2016; Martius & Lampert, 2016; Raissi, 2018; Cranmer et al., 2020a) related to the task, **(b)** a completely specified physics model (Raissi et al., 2017a; Raissi, 2018; Jiang et al., 2019) or with missing terms (Yin et al., 2021), and **(c)** different domain-specific physical constraints such as energy conservation (Greydanus et al., 2019; Cranmer et al., 2020b) or symmetries (Wang et al., 2020b; Finzi et al., 2021; Brandstetter et al., 2022a). While these PIML methods improve upon standard neural networks, damped pendulum results in Figure 3b show that they are generally not designed for OOD forecasting tasks. To study reasons for this failure, we categorize these methods into inductive and transductive based on their assumptions over ground-truth ODE parameters $\boldsymbol{W}^*$.

Transductive PIML methods. Transductive inference focuses on predicting missing parts from the observed data. In PIML, transductive inference methods focus on forecasting tasks with the same dynamical system parameter, and treat a test task with a different parameter as an unrelated task. For instance, SINDy (Brunton et al., 2016), EQL (Martius & Lampert, 2016), and related methods (Raissi, 2018; Chen, 2021), learn the ODE based on a dictionary of basis functions for a specific ground-truth ODE parameter $\boldsymbol{W}^{(i)*}$. These transductive methods (Figure 2b) do not transfer knowledge learnt in training to predict test examples with a different $\boldsymbol{W}^{(j)*}$. This forces these methods to forecast simply based on the initial observations of the test task alone, often leading to poor performance. Another class of transductive methods (Raissi et al., 2017a;b; Yu et al., 2022) assume that the ODE parameters $\boldsymbol{W}^*$ remain constant across all training and test tasks, and regularize neural networks to respect a given physics model. They have also been shown to be challenging to train for harder differential equations (Krishnapriyan et al., 2021) or return trivial solutions (Leiteritz & Pflüger, 2021). Causal PINNs (Wang et al., 2022) ensure that predictions at any time less than t are accurately resolved *before* predictions at time t, but do not allow for causal interventions to initial states and unknown parameters of the dynamical system. These methods will perform poorly if different training tasks have different ODE parameters.

Inductive PIML methods. Taking the opposite approach, *inductive inference* focuses on learning rules from the training data that can be applied to unseen test examples. Inductive methods dominate PIML approaches but are fragile OOD, since the learned rules are learned within the scope of the training data and are not guarantee to work outside the training data scope. For example, APHYNITY (Yin et al., 2021) and NDS (Mehta et al., 2021) are such inductive methods that augment a neural network to a known incomplete physics model where the parameters of the physics model are predicted inductively using a recurrent network. As illustrated in Figure 2c, these methods are able to learn from training tasks with different ODE parameters $\boldsymbol{W}^{(i)*}$. However, the recurrent network in APHYNITY fails OOD and often returns incorrect physics parameters OOD. Further, the augmented neural network suffers from the same extrapolation issues discussed in Section 3.1 leading to poor OOD performance as seen in Figure 3b. Concurrent work by Park et al. (2023) aligns

with our meta-learning architecture but lacks an IRM-type penalty that we empirically demonstrate as crucial for identifying the true causal structure.

With these key reasons identified for the fragility of existing methods to OOD scenarios, we propose an approach (*MetaPhysiCa*) that includes basis functions in the architecture for algorithmic alignment and can inductively transfer knowledge from varied training tasks to an OOD test task.

4 Proposed approach: MetaPhysiCa

We first describe a family of structural causal models describing dynamical systems (Section 4.1), then discuss the proposed architecture along with a causal structure discovery approach (Section 4.2) to learn the correct causal model for improved OOD performance.

4.1 Structural causal model

We describe the dynamical system using a deterministic structural causal model (Peters et al., 2022) with measurement noise over the observed states and explicitly define the assumptions over the unknown function ψ of Definition 1. The causal diagram is depicted in Figure 4 in the plated notation iterating over time $t = t_0, \ldots, t_{T^{(i)}}$ for each task i. As before, the hidden state of the dynamical system is $\boldsymbol{x}_t^{(i)} \in \mathbb{R}^d$ for task i. We define the causal process at each time step t for task i as follows.

Let $f_k(\cdot; \boldsymbol{\xi}_k) : \mathbb{R}^d \to \mathbb{R}, 1 \leq k \leq m$, be m linearly independent basis functions each with a separate set of parameters $\boldsymbol{\xi}_k$ acting on an input state $\boldsymbol{x}_t^{(i)}$. Examples of such basis functions include trigonometric functions like $f_1(\boldsymbol{x}_t^{(i)}; \boldsymbol{\xi}_1) = \sin(\xi_{1,1} x_{t,1}^{(i)} + \xi_{1,2})$, polynomial functions like $f_2(\boldsymbol{x}_t^{(i)}; \boldsymbol{\xi}_2) = x_{t,1}^{(i)} x_{t,2}^{(i)}$, and so on. The corresponding outputs from these basis are shown as $z_{k,t}^{(i)} := f_k(\boldsymbol{x}_t^{(i)}; \boldsymbol{\xi}_k)$ in Figure 4. The derivative $d\boldsymbol{x}_{t,j}^{(i)}/dt$ for a particular dimension $j \in \{1, \ldots, d\}$ is only affected by a few basis function outputs $z_{k,t}^{(i)}$ (green arrows in Figure 4). These selected basis functions along with their parameters $\boldsymbol{\xi}_k$ are shared across all the tasks $i = 1, \ldots, M$, i.e., the causal structure remains the same. The derivative is a linear combination of these selected basis functions with task-specific coefficients $\boldsymbol{W}^{(i)*}$ and dictates the next dynamical system state. We observe the dynamical system with independent measurement noise $\mathbf{X}_t^{(i)} := \boldsymbol{x}_t^{(i)} + \boldsymbol{\varepsilon}_t^{(i)}$, where $\boldsymbol{\varepsilon}_t^{(i)} \sim \mathcal{N}(\mathbf{0}, \sigma_\varepsilon^2 I)$.

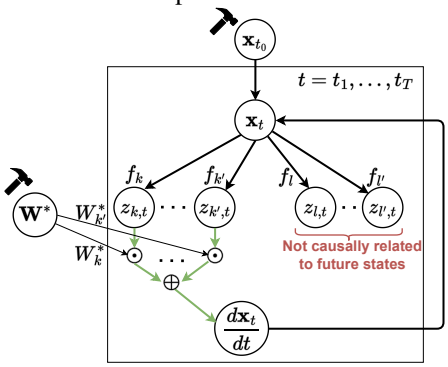


Figure 4: SCM for a dynamical system defined as an unknown linear combination of basis functions. We allow arbitrary interventions over initial conditions $\boldsymbol{x}_{t_0}$ and ODE parameters $\boldsymbol{W}^*$.

We assume that we are given the collection of m possible basis functions $f_k(\cdot; \boldsymbol{\xi}_k), k = 1, \ldots, m$, with unknown $\boldsymbol{\xi}_k$ and *no prior knowledge of which* $\{f_k\}_{k=1}^m$ *causally influence* $d\boldsymbol{x}_t^{(i)}/dt$. The need for basis functions stems from our analysis in Section 3.1, where we show that appropriate basis functions must be incorporated within the architecture in order to extrapolate to OOD scenarios.

4.2 Meta learning & model architecture

Given the training data $\{(\boldsymbol{x}_t^{(i)})_t\}_{i=1}^M$ generated from the unknown SCM described above, our goal is two-fold: **(a)** discover the true underlying causal structure, i.e., learn which basis functions f_k causally affect $d\boldsymbol{x}_{t,j}/dt$ for $j = 1, ..., d$, along with their parameters $\boldsymbol{\xi}_k$, and **(b)** learn the ground truth task-specific parameters $\boldsymbol{W}^{(i)*}$ that act as coefficients in linear combination of the selected basis functions. In the following, we propose a meta-learning framework that introduces structure parameters Φ that are shared across tasks and task-specific coefficients $\boldsymbol{W}^{(i)}$ that vary across the tasks:

$$\frac{d\hat{\mathbf{X}}_t^{(i)}}{dt} = (\boldsymbol{W}^{(i)} \odot \Phi) F(\hat{\mathbf{X}}_t^{(i)}; \boldsymbol{\xi}) \,, \tag{3}$$

where $\odot$ is the Hadamard product and

- $F(\hat{\mathbf{X}}_t^{(i)}; \boldsymbol{\xi}) := \begin{bmatrix} f_1(\hat{\mathbf{X}}_t^{(i)}; \boldsymbol{\xi}_1) & \cdots & f_m(\hat{\mathbf{X}}_t^{(i)}; \boldsymbol{\xi}_m) \end{bmatrix}^T$ is the vector of outputs from the basis functions with parameters $\boldsymbol{\xi} = \{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_m\}$,

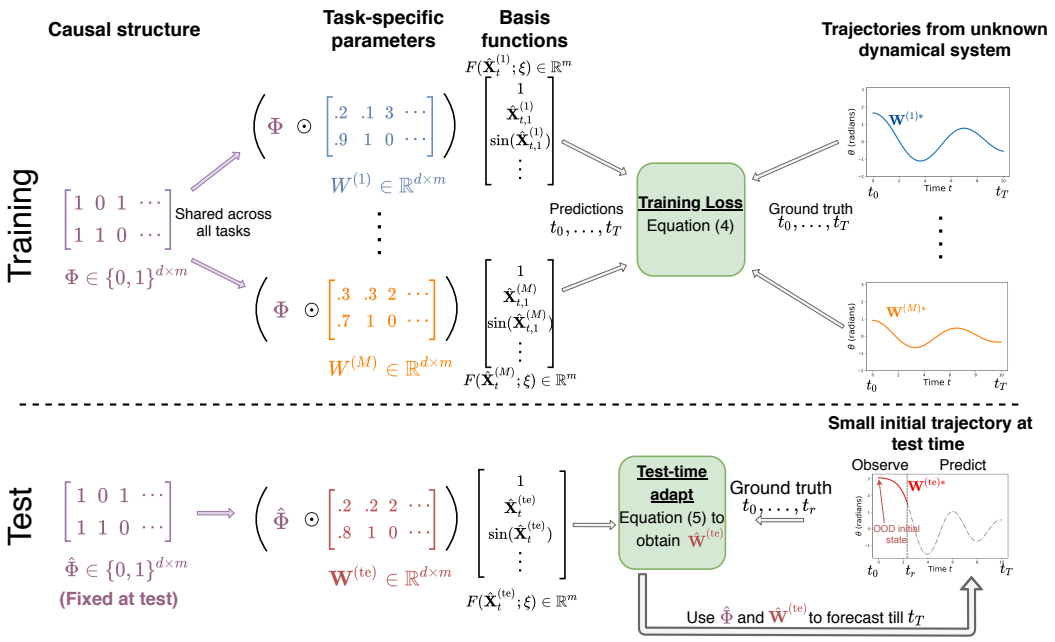


Figure 5: Schematic diagram of MetaPhysiCa and its training/test methodologies. We observe M trajectories in training from the same dynamical system with different initial conditions and ODE parameters. In training, Φ, denoting the causal structure, is shared among all tasks $i = 1, \ldots, M$, while $\boldsymbol{W}^{(i)}$ are the task-specific parameters, both learnt using Equation (4). During test, given small initial observations of the test trajectory from time $t_0, \ldots, t_r$, we use Equation (5) to adapt $\boldsymbol{W}^{(\text{te})}$ keeping the learnt causal structure $\hat{\Phi}$ fixed. Final forecast for the test trajectory for $t > t_r$ is given by solving the ODE in Equation (3) with optimal $\hat{\boldsymbol{W}}^{(\text{te})}$ and $\hat{\Phi}$.

- $\Phi \in \{0,1\}^{d \times m}$ are the learnable parameters governing the global causal structure across all tasks such that $\Phi_{j,k} = 1$ iff edge $z_{k,t} \to {}^{d\boldsymbol{x}_{t,j}}/_{dt}$ exists in Figure 4,
- $\boldsymbol{W}^{(i)} \in \mathbb{R}^{d \times m}$ are task-specific parameters that act as coefficients in a linear combination of the selected basis functions.

Figure 5 shows a schematic diagram of MetaPhysiCa along with the training/test methodologies (described next). Additional implementation details are presented in Appendix C.

Next we describe a procedure to learn the structure parameters Φ. Finding whether an edge exists or not in the causal graph is known as the causal structure discovery problem (e.g., Heinze-Deml et al. (2018)). We use a score-based causal discovery approach (e.g., Huang et al. (2018)) where we assign a score to each possible causal graph. We wish to find the *minimal* causal structure, i.e., with the least number of edges, that also fits the training data. This balances the complexity of the causal structure with training likelihood, and avoids overfitting the training data.

Prediction loss & sparse structure. A sparse structure for Φ implies fewer terms in the RHS of the learnt equation for the derivatives in Equation (3). Several causal discovery approaches have been proposed that learn such minimal causal structure via continuous optimization (Zheng et al., 2018; Ng et al., 2022). We use the log-likelihood of the training data with ℓ_1-regularization term to induce sparsity that is known to perform well for general causal structure discovery tasks (Zheng et al., 2018). Note that since the direction of all the edges are known (i.e., $z_{k,t} \to {}^{d\boldsymbol{x}_{t,j}}/_{dt}$), we do not need the acyclicity constraints. and the causal graph is uniquely identified by its Markov equivalence class (Pearl, 2009, Chapter 2).

The prediction error is given by $R^{(i)}(\boldsymbol{W}^{(i)}, \Phi, \boldsymbol{\xi}) := \frac{1}{T^{(i)}+1} \sum_{t=t_0}^{t_{T^{(i)}}} ||\hat{\mathbf{X}}_t^{(i)} - \mathbf{X}_t^{(i)}||_2^2$ where $\hat{\mathbf{X}}_t^{(i)} = \mathbf{X}_{t_0}^{(i)} + \int_{t_0}^{t} (\boldsymbol{W}^{(i)} \odot \Phi) F(\hat{\mathbf{X}}_\tau^{(i)}; \boldsymbol{\xi}) d\tau$ are the predictions obtained using an ODE solver to integrate Equation (3). In practice however, we found the squared loss directly between the predicted and estimated ground truth derivatives, i.e., $\widetilde{R}^{(i)}(\boldsymbol{W}^{(i)}, \Phi, \boldsymbol{\xi}) = \frac{1}{T^{(i)}+1} \sum_{t=t_0}^{t_{T^{(i)}}} ||{}^{d\hat{\mathbf{X}}_t^{(i)}}/_{dt} - {}^{d\mathbf{X}^{(i)}}/_{dt}||_2^2$, leads to a stable learning procedure with better accuracy in-distribution and OOD. As discussed before, we use an ℓ_1-regularization term $||\Phi||_1$ to learn a causal structure with the fewest possible edges $z_{k,t} \to {}^{d\boldsymbol{x}_{t,j}}/_{dt}, j = 1, \ldots, d$, while minimizing the prediction error in training.

Learning Φ that induces equal risk across all tasks. Our structure discovery task comes with an additional challenge as each training task is obtained with potentially different initial condition $\boldsymbol{x}_{t_0}^{(i)}$ and ODE parameter $\boldsymbol{W}^{(i)*}$. Existing causal discovery approaches via continuous optimization (e.g., (Zheng et al., 2018)) may not learn the correct structure. For example, if the (unknown) ground-truth weight for a basis function f_k is nonzero only for a small percentage of training tasks, standard approaches may learn a structure without this basis function (i.e., sacrificing accuracy for a few training tasks to learn a simpler model). This will impact OOD performance if many of the test tasks now have (unknown) nonzero ground-truth weight for the basis function f_k.

Our goal then is to learn a structure that minimizes the prediction error across **all** training tasks simultaneously, similar to learning robust representations via invariant risk minimization-type methods (Arjovsky et al., 2019; Krueger et al., 2021). We use a V-REx regularization (Krueger et al., 2021) that minimizes the variance of prediction errors across tasks. We empirically show in an ablation study (Table 7 in Appendix D.2) that V-REx penalty is necessary to learn the true causal structure.

Meta-learning objective. Now we are ready to describe our final optimization objective. Similar to standard meta-learning objectives (Finn et al., 2017; Franceschi et al., 2018; Hospedales et al., 2021), we propose a bi-level objective that optimizes the structure parameters Φ and the global parameters ξ in the outer-level, and the task-specific parameters $\boldsymbol{W}^{(i)}$ in the inner-level as follows

$$\hat{\Phi}, \hat{\boldsymbol{\xi}} = \arg\min_{\Phi, \boldsymbol{\xi}} \frac{1}{M} \sum_{i=1}^{M} R^{(i)}(\hat{\boldsymbol{W}}^{(i)}, \Phi, \boldsymbol{\xi}) + \lambda_{\Phi} ||\Phi||_1 + \lambda_{\text{REx}} \text{Variance}(\{R^{(i)}(\hat{\boldsymbol{W}}^{(i)}, \Phi, \boldsymbol{\xi})\}_{i=1}^{M})$$
$$\text{s.t.} \forall i, \hat{\boldsymbol{W}}^{(i)} = \arg\min_{\boldsymbol{W}^{(i)}} R^{(i)}(\boldsymbol{W}^{(i)}, \Phi, \boldsymbol{\xi}) \,, \tag{4}$$

where λ_{Φ} and λ_{REx} are hyperparameters. We approximate the discrete structure parameters Φ using deterministic binarization techniques (Courbariaux et al., 2015; 2016). We reparameterize $\Phi_{j,k} := \mathbf{1}(\sigma(\widetilde{\Phi}_{j,k}) > 0.5)$ where $\Phi' \in \mathbb{R}^{d \times m}$, $\sigma(\cdot)$ is the sigmoid function, and the gradients are estimated via straight-through-estimator. The bi-level optimization in Equation (4) can be approximated by alternate optimization steps for $(\Phi, \boldsymbol{\xi})$ and $\{\boldsymbol{W}^{(i)}\}_{i=1}^{M}$ in outer and inner loops respectively (Borkar, 1997; Chen et al., 2021). In our experiments, jointly optimizing all parameters instead resulted in comparable performance with considerable computational benefits (Appendix D.2.3). We choose the hyperparameters λ_{Φ} and λ_{REx} that result in sparsest model (i.e., with the least $||\hat{\Phi}||_0$) while achieving validation loss within 5% of the best *in-distribution* validation loss. Use of *in-distribution data for validation* is a key requirement for OOD tasks as we do not have access to the test distribution.

In what follows, we show that MetaPhysiCa learns the true causal structure in Figure 4 given a set of assumptions described in Assumption 1 (Appendix A).

Theorem 1 (MetaPhysiCa identifies the true causal structure)**.** *Under Assumption 1 (Appendix A) there exists a $\lambda_{\Phi} > 0$ in the optimization objective of Equation* (4) *such that the optimal $\hat{\Phi}$ learns the true causal structure in the SCM of Figure 4, i.e., $\hat{\Phi}_{j,k}$ = 1 iff there exists an edge $z_{k,t} \rightarrow {}^{d\boldsymbol{x}_{t,j}}/_{dt}$ in the true causal graph.*

Proof in Appendix A shows that, given a small enough $\lambda_{\Phi} > 0$, the true causal structure Φ^* is the unique minimizer of the objective in Equation (4). While Theorem 1 holds only for SCM in Figure 4, MetaPhysiCa can be extended to more expressive SCMs by composing basis functions (e.g., to obtain $\sin(x_{t,1}^2)$). MetaPhysiCa with such an expressive SCM shows OOD performance gains on a complex ODE task (Appendix D.4), but can suffer from learning stiff ODEs due to the complexity of such a 2-layer composition procedure. Better optimization techniques may help alleviate this problem.

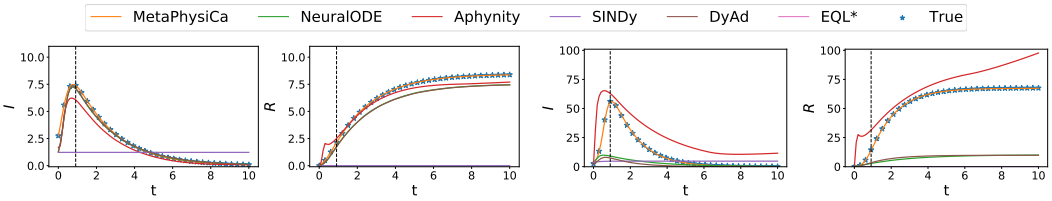
4.3 Test-time adaptation

After learning the optimal causal structure $\hat{\Phi}$, we adapt the model's task-specific parameters for each test task using the initial test observations available up to time t_r. The test task may involve (unknown) out-of-support ground-truth parameters $\boldsymbol{W}^{(\text{te})*}$ and out-of-support initial conditions $\boldsymbol{x}_{t_0}^{(\text{te})}$. Keeping $\hat{\Phi}, \hat{\boldsymbol{\xi}}$ fixed, we optimize the following to adapt the model's task-specific parameters $\boldsymbol{W}^{(\text{te})}$,

$$\hat{\boldsymbol{W}}^{(\text{te})} = \arg\min_{\boldsymbol{W}^{(\text{te})}} \frac{1}{t_r + 1} \sum_{t=t_0}^{t_r} ||\hat{\mathbf{X}}_t^{(\text{te})} - \mathbf{X}_t^{(\text{te})}||_2^2 \,, \tag{5}$$

Table 1: **(Epidemic model results)** Test NRMSE ↓ for different methods. NaN* indicates that the model returned errors during test. **MetaPhysiCa outputs** $28\times$ **and** $9\times$ **more robust OOD predictions** for the two OOD scenarios respectively.

Methods	Test Normalized RMSE (NRMSE) ↓ ID	OOD $\mathbf{X}_{t_0}$	OOD $\mathbf{X}_{t_0}$ and $\boldsymbol{W}^*$
Standard Deep Learning			
NeuralODE (Chen et al., 2018)	0.005 (0.000)	1.139 (0.031)	1.073 (0.102)
Meta Learning			
DyAd (Wang et al., 2021b)	0.006 (0.001)	1.147 (0.044)	1.207 (0.202)
CoDA (Kirchmeyer et al., 2022)	0.004 (0.001)	1.341 (0.389)	1.090 (0.274)
Physics-informed Machine Learning			
APHYNITY (Yin et al., 2021)	0.151 (0.150)	0.544 (0.249)	0.898 (0.211)
SINDy (Brunton et al., 2016)	1.999 (0.046)	2.746 (0.476)	NaN*
EQL (Martius & Lampert, 2016)	NaN*	NaN*	NaN*
MetaPhysiCa **(Ours)**	0.009 (0.004)	**0.019 (0.002)**	**0.100 (0.080)**



(a) In-distribution predictions (b) Predictions under OOD $\mathbf{X}_{t_0}$

Figure 6: **(Epidemic model results)** Ground truth curves (blue stars) in- and out-of-distribution along with corresponding predictions. **Only MetaPhysiCa (orange) closely follows the true curve OOD.**

where $\hat{\mathbf{X}}_t^{(\text{te})} = \mathbf{X}_{t_0}^{(\text{te})} + \int_{t_0}^{t} (\boldsymbol{W}^{(\text{te})} \odot \hat{\Phi}) F(\hat{\mathbf{X}}_\tau^{(\text{te})}; \hat{\boldsymbol{\xi}}) d\tau$ are the predictions obtained using the optimal structure $\hat{\Phi}$. The final predictions for $t > t_r$ are obtained with the adapted parameters $\hat{\boldsymbol{W}}^{(\text{te})}$ and the fixed global parameters $\hat{\Phi}, \hat{\boldsymbol{\xi}}$ (see Figure 5 bottom).

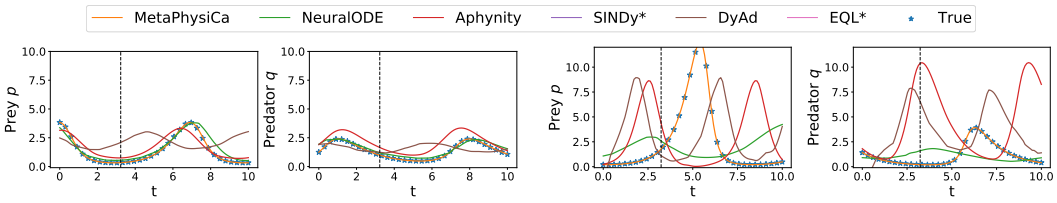
Two key aspects of the test-time adaptation: **(a)** Only the task-specific model parameters $\boldsymbol{W}^{(\text{te})}$ are adapted whereas the meta-model $\hat{\Phi}$ learnt during training is kept fixed, and **(b)** only the observations from time $t_0, \ldots, t_r$ of the given test trajectory is used to adapt the parameters $\boldsymbol{W}^{(\text{te})}$. This allows the model to be robust to out-of-distribution ground-truth ODE parameters $\boldsymbol{W}^{(\text{te})*}$.

5 Empirical Evaluation

We evaluate **MetaPhysiCa**[1] in synthetic forecasting tasks based on 3 different dynamical systems (ODEs) from the literature (Yin et al., 2021; Wang et al., 2021a) adapted to our OOD scenario, namely, **(i)** Damped pendulum system, **(ii)** Predator-prey system and **(iii)** Epidemic model. We compare against: **(a) NeuralODE** (Chen et al., 2018), a deep learning method for learning ODEs, **(b) DyAd** (Wang et al., 2021b) (modified for ODEs), that adapts to different training tasks with a weakly-supervised encoder, **(c) CoDA** (Kirchmeyer et al., 2022), that learns to modify its parameters to each task with a low-rank adaptation, **(d) APHYNITY** (Yin et al., 2021), that augments a known incomplete physics model with a neural network, **(e) SINDy** (Brunton et al., 2016), a *transductive* PIML method that uses sparse regression to learn linear coefficients over a given set of basis functions, **(f) EQL** (Martius & Lampert, 2016), a *transductive* PIML method that uses basis functions within a neural network and learns a sparse model (details on all models in Appendix C).

Dataset generation. For each dynamical system, we simulate the respective ODE as per Definition 1 to generate $M = 1000$ training tasks each observed over regularly-spaced discrete time steps $t = 0, ..., t_T$ with $\Delta t = 0.1$. For each training task i, we sample an initial condition $\boldsymbol{x}_{t_0}^{(i)} \sim P^{\text{tr}}(\boldsymbol{x}_{t_0})$ and $\boldsymbol{W}^{(i)*} \sim P^{\text{tr}}(\boldsymbol{W}^*)$. At OOD test, we generate $M' = 200$ test tasks by simulating the dynamical system over timesteps $t = 0, ..., t_r$ with $\Delta t = 0.1$ with $r = T/3$ or $T/10$. For every test task j, we set initial conditions $\boldsymbol{x}_{t_0}^{(j)}$ and test ODE parameters $\boldsymbol{W}^{(j)*}$ outside their respective training supports.

[1]Code is available at https://github.com/PurdueMINDS/MetaPhysiCa



(a) In-distribution predictions

(b) Predictions under OOD $\mathbf{X}_{t_0}$

Figure 7: **(Predator-prey results)** Ground truth curves (blue stars) in- and out-of-distribution along with corresponding predictions. **Only MetaPhysiCa (orange) closely follows the true curve OOD.**

We consider two OOD scenarios: **(a) OOD $\mathbf{X}_{t_0}$**, when only the initial conditions are out-of-support, and **(b) OOD $\mathbf{X}_{t_0}$ and $\boldsymbol{W}^*$**, when both initial conditions and ODE parameters are out-of-support.

We consider three dynamical systems in our experiments, with 3 to 6 RHS terms in their respective differential equations: a damped pendulum system (Yin et al., 2021), a predator-prey system (Wang et al., 2021a), and an epidemic (SIR) model (Wang et al., 2021a), with following OOD shifts in their initial conditions respectively: acute initial angles in training to nearly vertical initial angles in OOD test, initial prey population $10\times$ less in OOD test than in training, and initial population susceptible to a disease $10\times$ more in OOD test than in training. For all three dynamical systems, all ODE parameters are $\approx 1.5\times$ higher OOD than in training (with non-overlapping support). We generate the damped pendulum dataset with 1% zero-mean Gaussian noise and the rest with no noise to show that OOD failure of baselines is unrelated to noise: existing methods fail OOD even with clean observations. Detailed description of the datasets is presented in Appendix B.

Results. We report normalized root mean squared errors (i.e., RMSE normalized with ground truth standard deviation) for the in-distribution (ID), OOD w.r.t. $\mathbf{X}_{t_0}$, and OOD w.r.t. $\mathbf{X}_{t_0}$ and $\boldsymbol{W}^*$ in Tables 1, 2 and 3 (latter two in appendix) for the 3 datasets. Figures 3, 6 and 7 show example predictions in- and out-of-distribution from all models. NeuralODE, DyAd, CoDA and APHYNITY use neural network components and are able to learn the in-distribution task well with low errors. However, the corresponding errors OOD are high as they are unable to adapt to OOD initial conditions and OOD parameters. Example OOD predictions (Figures 3b, 6b and 7b) from these methods show that they have not learnt the true dynamics of the system. For example, for epidemic modeling (Figure 6b), most models predict trajectories very similar to training trajectories even though the number of susceptible individuals is $10\times$ higher in OOD test. SINDy and EQL, being transductive, cannot use the training data and are fit on the initial test observations alone. Thus, they are unable to identify an accurate analytical equation from these few test observations, resulting in prediction issues due to stiff ODEs. MetaPhysiCa performs the best OOD across all datasets achieving $2\times$ to $28\times$ lower OOD NRMSE than the best baseline, and closely follows the true curve OOD.

Qualitative analysis. Appendix D.1 (Table 5) shows that MetaPhysiCa's performance gains stem from two factors: **(i)** The optimal meta-model $\hat{\Phi}$ learns the ground truth ODE for all 3 dynamical systems, and **(ii)** the model adapts its task-specific parameters separately to each OOD test task. The former is key for robustness over OOD initial states and the latter helps to be robust over OOD parameters $\boldsymbol{W}^*$. Appendix D.2.1 shows ℓ_1-regularization and test-time adaptation are necessary components; OOD performance degrades significantly without either. Appendix D.2.2 shows that V-REx penalty is necessary to learn the true causal structure, without which OOD NRMSE is $24\times$ worse. Appendix D.7 shows that MetaPhysiCa is able to learn the true dynamics for varying number of basis functions, and Appendix D.6 evaluates MetaPhysiCa without full algorithmic alignment.

6 Conclusions

We considered the out-of-distribution task of forecasting a dynamical system (ODE) under new initial conditions and new ODE parameters. We showed that existing PIML methods do not perform well in these tasks and proposed MetaPhysiCa that uses a meta-learning framework to learn the causal structure for the shared dynamics across all tasks, while adapting the task-specific parameters. Results on three OOD forecasting tasks show that MetaPhysiCa is more robust with $2\times$ to $28\times$ reduction in OOD error compared to the best baseline. **Limitations & future work:** We believe that forecasting models should be robust to OOD shifts, and that our work takes a step in the right direction with several potential avenues for future research: **(i)** Extending MetaPhysiCa to forecasting PDEs under OOD scenarios requires an expanded set of basis functions that includes differential operators, and considering OOD boundary conditions. **(ii)** Better optimization techniques to avoid learning stiff ODEs when extending MetaPhysiCa to more expressive SCMs.

ACKNOWLEDGMENTS

This work was funded in part by the National Science Foundation (NSF) awards, CCF-1918483, CAREER IIS-1943364 and CNS-2212160, Amazon Research Award, AnalytiXIN, and the Wabash Heartland Innovation Network (WHIN), Ford, Nvidia, CISCO, and Amazon. Computing infrastructure was supported in part by CNS-1925001 (CloudBank). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

REFERENCES

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Vivek S Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5): 291–294, 1997.

Johannes Brandstetter, Max Welling, and Daniel E Worrall. Lie point symmetry data augmentation for neural pde solvers. *arXiv preprint arXiv:2202.07643*, 2022a.

Johannes Brandstetter, Daniel E Worrall, and Max Welling. Message passing neural pde solvers. In *International Conference on Learning Representations*, 2022b.

Steven L. Brunton, Joshua L. Proctor, J. Nathan Kutz, and William Bialek. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences of the United States of America*, 113(15):3932–3937, 2016. ISSN 10916490. doi: 10.1073/pnas.1517384113.

Rick Chartrand. Numerical differentiation of noisy, nonsmooth data. *International Scholarly Research Notices*, 2011, 2011.

Gang Chen. Learning symbolic expressions via gumbel-max equation learner networks. *arXiv preprint arXiv:2012.06921*, 2020.

Gang Chen. Learning Symbolic Expressions via Gumbel-Max Equation Learner Networks. *arXiv:2012.06921 [cs]*, May 2021.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 6572–6583, Red Hook, NY, USA, December 2018. Curran Associates Inc.

Tianyi Chen, Yuejiao Sun, and Wotao Yin. Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems. *Advances in Neural Information Processing Systems*, 34: 25294–25307, 2021.

Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. *Advances in neural information processing systems*, 28, 2015.

Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016.

M. Cranmer, Alvaro Sanchez-Gonzalez, P. Battaglia, Rui Xu, K. Cranmer, D. Spergel, and S. Ho. Discovering Symbolic Models from Deep Learning with Inductive Biases. *NeurIPS*, 2020a.

Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020b.

Yuchen Dang, Zheyuan Hu, Miles Cranmer, Michael Eickenberg, and Shirley Ho. Tnt: Vision transformer for turbulence simulations. *arXiv preprint arXiv:2207.04616*, 2022.

Arka Daw, Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*, 2017.

James H Faghmous and Vipin Kumar. A big data guide to understanding climate change: The case for theory-guided data science. *Big data*, 2(3):155–163, 2014.

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*, pp. 1126–1135. PMLR, July 2017.

Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In *International Conference on Machine Learning*, pp. 3318–3328. PMLR, 2021.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pp. 1568–1577. PMLR, 2018.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.

Christina Heinze-Deml, Marloes H Maathuis, and Nicolai Meinshausen. Causal structure learning. *Annual Review of Statistics and Its Application*, 5:371–391, 2018.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9): 5149–5169, 2021.

Biwei Huang, Kun Zhang, Yizhu Lin, Bernhard Schölkopf, and Clark Glymour. Generalized Score Functions for Causal Discovery. *KDD : proceedings. International Conference on Knowledge Discovery & Data Mining*, 2018:1551–1560, August 2018. ISSN 2154-817X. doi: 10.1145/3219819.3220104.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510, 2017.

Chiyu "Max" Jiang, Karthik Kashinath, Prabhat, and Philip Marcus. Enforcing Physical Constraints in Neural Neural Networks through Differentiable PDE Layer. September 2019.

Matthieu Kirchmeyer, Yuan Yin, Jeremie Dona, Nicolas Baskiotis, Alain Rakotomamonjy, and Patrick Gallinari. Generalizing to New Physical Systems via Context-Informed Dynamics Model. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 11283–11301. PMLR, June 2022.

Dmitrii Kochkov, Jamie A Smith, Ayya Alieva, Qing Wang, Michael P Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21), 2021.

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.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pp. 5815–5826. PMLR, 2021.

Raphael Leiteritz and Dirk Pflüger. How to Avoid Trivial Solutions in Physics-Informed Neural Networks. *arXiv:2112.05620 [cs, stat]*, December 2021.

Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, November 2016. ISSN 0022-1120, 1469-7645. doi: 10.1017/jfm.2016.615.

Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):1–10, 2018.

Georg Martius and Christoph H. Lampert. Extrapolation and learning equations. *arXiv:1610.02995 [cs]*, October 2016.

Viraj Mehta, Ian Char, Willie Neiswanger, Youngseog Chung, Andrew Nelson, Mark Boyer, Egemen Kolemen, and Jeff Schneider. Neural dynamical systems: Balancing structure and flexibility in physical prediction. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 3735–3742. IEEE, 2021.

Ignavier Ng, Shengyu Zhu, Zhuangyan Fang, Haoyang Li, Zhitang Chen, and Jun Wang. Masked gradient-based causal structure learning. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pp. 424–432. SIAM, 2022.

MoonJeong Park, Youngbin Choi, Namhoon Lee, and Dongwoo Kim. Spreme: Sparse regression for multi-environment dynamic systems. In *When Machine Learning meets Dynamical Systems: Theory and Applications workshop, AAAI*, 2023.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Jonas Peters, Stefan Bauer, and Niklas Pfister. Causal models for dynamical systems. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, pp. 671–690. 2022.

Maziar Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *The Journal of Machine Learning Research*, 19(1):932–955, 2018.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017a.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10566*, 2017b.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Multistep neural networks for data-driven discovery of nonlinear dynamical systems. *arXiv preprint arXiv:1801.01236*, 2018.

Paul K Rubenstein, Stephan Bongers, Bernhard Schölkopf, and Joris M Mooij. From deterministic odes to dynamic structural causal models. *arXiv preprint arXiv:1608.08028*, 2016.

Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 2009.

Sebastian Thrun and Lorien Pratt. Learning to learn: Introduction and overview. In *Learning to learn*, pp. 3–17. Springer, 1998.

Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. Towards physics-informed deep learning for turbulent flow prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1457–1466, 2020a.

Rui Wang, Robin Walters, and Rose Yu. Incorporating symmetry into deep dynamics models for improved generalization. *arXiv preprint arXiv:2002.03061*, 2020b.

Rui Wang, Danielle Maddix, Christos Faloutsos, Yuyang Wang, and Rose Yu. Bridging physics-based and data-driven modeling for learning dynamical systems. In *Learning for Dynamics and Control*, pp. 385–398. PMLR, 2021a.

Rui Wang, Robin Walters, and Rose Yu. Meta-learning dynamics forecasting using task inference. *arXiv preprint arXiv:2102.10271*, 2021b.

Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality is all you need for training physics-informed neural networks. *arXiv preprint arXiv:2203.07404*, 2022.

Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919*, 2020.

SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pp. 802–810, 2015.

Keyulu Xu, Mozhi Zhang, Jingling Li, Simon Shaolei Du, Ken-Ichi Kawarabayashi, and Stefanie Jegelka. How neural networks extrapolate: From feedforward to graph neural networks. In *International Conference on Learning Representations*, 2021.

Alireza Yazdani, Lu Lu, Maziar Raissi, and George Em Karniadakis. Systems biology informed deep learning for inferring parameters and hidden dynamics. *PLoS computational biology*, 16(11): e1007575, 2020.

Kyongmin Yeo and Igor Melnyk. Deep learning algorithm for data-driven simulation of noisy dynamical system. *Journal of Computational Physics*, 376:1212–1231, January 2019. ISSN 00219991. doi: 10.1016/j.jcp.2018.10.024.

Yuan Yin, Le Vincent, DONA Jérémie, Emmanuel de Bezenac, Ibrahim Ayed, Nicolas Thome, et al. Augmenting physical models with deep networks for complex dynamics forecasting. In *International Conference on Learning Representations*, 2021.

Jeremy Yu, Lu Lu, Xuhui Meng, and George Em Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse pde problems. *Computer Methods in Applied Mechanics and Engineering*, 393:114823, 2022.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Supplementary Material of "MetaPhysiCa: Improving OOD Robustness in Physics-informed Machine Learning"

A PROOF OF THEOREM

Assumption 1. *We are given M tasks generated from the structural causal model in Figure 4 with infinite observations per task, i.e., $T^{(i)} \to \infty$, $\forall i \in \{1, \dots, M\}$. Then, for all $p \in \{1, \dots, d\}$ and every basis function f_k active in the true (unknown) casual model with $\Phi^*_{p,k} = 1$, we assume there exists $\nu > 0$ and at least one task $j \in \{1, \dots, M\}$ such that,*

- $|\boldsymbol{W}^{(j)*}_{p,k} f_k(\boldsymbol{x}^{(j)}_t; \boldsymbol{\xi})| > \nu$ *for all t, and*
- f_k *is linearly independent from all other basis functions in the domain of task j,*

where $\boldsymbol{W}^{(j)*}_{p,k}$ is the (unknown) ground truth parameter corresponding to the basis function f_k to predict ${}^{d\boldsymbol{x}^{(j)}_{t,p}}/_{dt}$.

Theorem 1 (MetaPhysiCa identifies the true causal structure)**.** *Under Assumption 1 (Appendix A) there exists a $\lambda_\Phi > 0$ in the optimization objective of Equation* (4) *such that the optimal $\hat{\Phi}$ learns the true causal structure in the SCM of Figure 4, i.e., $\hat{\Phi}_{j,k} = 1$ iff there exists an edge $z_{k,t} \to {}^{d\boldsymbol{x}_{t,j}}/_{dt}$ in the true causal graph.*

Proof. In what follows, we prove the statement for 1-dimensional trajectories, i.e., $\mathbf{X}_t \in \mathbb{R}^d, d = 1$. The proof can be trivially extended for $d > 1$. Our optimization objective is

$$\hat{\Phi} = \arg\min_{\Phi} \mathcal{L}(\Phi) = \arg\min_{\Phi} \frac{1}{M} \sum_{i=1}^{M} \widetilde{R}^{(i)}(\hat{\boldsymbol{W}}^{(i)}(\Phi), \Phi, \boldsymbol{\xi}) + \lambda_\Phi ||\Phi||_1 \,, \tag{6}$$

where

- $\hat{\boldsymbol{W}}^{(i)}(\Phi) := \arg\min_{\boldsymbol{W}^{(i)}} \widetilde{R}^{(i)}(\boldsymbol{W}^{(i)}, \Phi, \boldsymbol{\xi})$ denotes the optimal task-specific parameter given a structure Φ. We use the notation $\hat{\boldsymbol{W}}^{(i)}(\Phi)$ to explicitly denote the dependence of the optimal task-specific parameter on a given structure Φ.
- $\widetilde{R}^{(i)}(\hat{\boldsymbol{W}}^{(i)}(\Phi), \Phi, \boldsymbol{\xi}) = \frac{1}{T} \sum_{t=t_1}^{t_T} ||{}^{d\hat{\mathbf{X}}^{(i)}_t}/_{dt} - {}^{d\mathbf{X}^{(i)}}/_{dt}||_2^2$, with the predictions from the proposed model given by $\frac{d\hat{\mathbf{X}}^{(i)}_t}{dt} = (\hat{\boldsymbol{W}}^{(i)}(\Phi) \odot \Phi) F(\hat{\mathbf{X}}^{(i)}_t; \boldsymbol{\xi})$.
- We assume that the parameters $\boldsymbol{\xi}$ of the basis functions are known.
- We set $\lambda_{\text{REx}} = 0$.

Let the true unknown causal structure be Φ^*. Note that from Assumption 1, Φ^* has the property that $\Phi^*_{1,k} = 1$ if and only if there exists a task $j \in \{1, \dots, M\}$ such that the ground truth parameters $\boldsymbol{W}^{(j)*}_k \neq 0$, otherwise $\Phi^*_{1,k} = 0$.

The inner optimization $\hat{\boldsymbol{W}}^{(i)}(\Phi) := \arg\min_{\boldsymbol{W}^{(i)}} \widetilde{R}^{(i)}(\boldsymbol{W}^{(i)}, \Phi, \boldsymbol{\xi})$ is a linear regression problem. Given the data for task i, $(\boldsymbol{x}^{(i)}_t)_{t=t_0}^{t_{T^{(i)}}}$, we can define an $T^{(i)} \times m$ design matrix $\mathbf{F}^{(i)}$ such that $\mathbf{F}^{(i)}_{t,k} := f_k(\boldsymbol{x}^{(i)}_t; \xi_k)$ for $t = t_1, \dots, t_{T^{(i)}}$. Also, we define $T^{(i)}$-length vector $\boldsymbol{y}^{(i)}$ of noisy ground truth derivatives where $\boldsymbol{y}^{(i)}_t := d\boldsymbol{x}^{(i)}_t/dt + \boldsymbol{\epsilon}_t$, for $t = t_1, \dots, t^{(i)}_T$, and $\boldsymbol{\epsilon}_t$ is a zero-mean Gaussian noise related to the noise in the ground truth observations. Then, with $T^{(i)} \to \infty$, $\hat{\boldsymbol{W}}^{(i)}(\Phi^*) \to \boldsymbol{W}^{(i)*}$ for all $i \in \{1, \dots, M\}$, i.e., if we knew the true causal structure, the corresponding task-specific parameters converge to the ground truth parameters $\boldsymbol{W}^{(i)*}$.

Assume $\hat{\Phi} \neq \Phi^*$, i.e., the optimal structure found in Equation (6) is not the true causal structure. In the following, we will show that this leads to a contradiction. For any Φ, define the function $b(\Phi) := \{f_k \mid \Phi_{1,k} = 1\}$ denoting the set of basis functions that are active in Φ. Consider $b(\hat{\Phi})$ and

$b(\Phi^*)$. $\mathcal{K} := b(\hat{\Phi}) \cap b(\Phi^*)$ denotes the basis functions that are shared by the true causal structure and $\hat{\Phi}$. Then essentially, the basis functions in $b(\Phi^*) \setminus \mathcal{K}$ are replaced by $b(\hat{\Phi}) \setminus \mathcal{K}$ (at least one of these sets is non-empty).

Case 1: $|b(\Phi^*) \setminus \mathcal{K}| = 0$, i.e., all basis functions in the true causal structure are present in $\hat{\Phi}$ but $\hat{\Phi}$ contains at least one additional basis function.

Then, with $T^{(i)} \to \infty$, the optimal error $\widetilde{R}^{(i)}(\hat{\boldsymbol{W}}^{(i)}(\hat{\Phi}), \hat{\Phi}, \boldsymbol{\xi}) \geq \widetilde{R}^{(i)}(\hat{\boldsymbol{W}}^{(i)}(\Phi^*), \Phi^*, \boldsymbol{\xi})$ with equality achieved, for example, when for all $k \in \mathcal{K}$, we have $\hat{\boldsymbol{W}}^{(i)}(\hat{\Phi})_{1,k} = \hat{\boldsymbol{W}}^{(i)}(\Phi^*)_{1,k}$ and for all $k' \in b(\hat{\Phi}) \setminus \mathcal{K}$, we have $\hat{\boldsymbol{W}}^{(i)}(\hat{\Phi})_{1,k'} = 0$. Furthermore, $||\hat{\Phi}||_1 - ||\Phi^*||_1 = |b(\hat{\Phi}) \setminus \mathcal{K}| - |b(\Phi^*) \setminus \mathcal{K}| = |b(\hat{\Phi}) \setminus \mathcal{K}| > 0$.

Thus, for any $\lambda_\Phi > 0$,

$$\mathcal{L}(\hat{\Phi}) = \frac{1}{M} \sum_{i=1}^{M} \widetilde{R}^{(i)}(\hat{\boldsymbol{W}}^{(i)}(\hat{\Phi}), \hat{\Phi}, \boldsymbol{\xi}) + \lambda_\Phi ||\hat{\Phi}||_1 > \frac{1}{M} \sum_{i=1}^{M} \widetilde{R}^{(i)}(\hat{\boldsymbol{W}}^{(i)}(\Phi^*), \Phi^*, \boldsymbol{\xi}) + \lambda_\Phi ||\Phi^*||_1 = \mathcal{L}(\Phi^*) \,.$$

This is a contradiction as $\hat{\Phi}$ is the optimal solution of Equation (6).

Case 2: $|b(\Phi^*) \setminus \mathcal{K}| > 0$. There exists a task $j \in \{1, \ldots, M\}$ such that

$$(\hat{\boldsymbol{W}}^{(j)}(\hat{\Phi}) \odot \hat{\Phi}) \, \mathbf{F}^{(j)} \neq (\hat{\boldsymbol{W}}^{(j)}(\Phi^*) \odot \Phi^*) \, \mathbf{F}^{(j)} \,, \tag{7}$$

where $\mathbf{F}^{(j)}$ is the design matrix corresponding to task j (defined above), $|\boldsymbol{W}_{1,k}^{(j)*} f_k(\boldsymbol{x}_t; \boldsymbol{\xi})| > \nu > 0$ for some $k \in b(\Phi^*) \setminus \mathcal{K}$, and f_k cannot be written as a linear combination of other basis functions in the domain of task j. There always exists at least one such task j, otherwise Assumption 1 is violated. Let $\eta_t^{(j)}$ be the difference between LHS and RHS of Equation (7), then $|\eta_t^{(j)}| > \nu'$ for some fixed $\nu' > 0$ and all t.

For this task j, the optimal error as $T^{(j)} \to \infty$,

$$\begin{aligned}
\widetilde{R}^{(j)}(\hat{\boldsymbol{W}}^{(j)}(\hat{\Phi}), \hat{\Phi}) &= \mathbb{E}_t \left[\left((\hat{\boldsymbol{W}}^{(j)}(\hat{\Phi}) \odot \hat{\Phi}) \, \mathbf{F}^{(j)} - y_t - \epsilon_t \right)^2 \right] \\
&= \mathbb{E}_t \left[\left((\hat{\boldsymbol{W}}^{(j)}(\Phi^*) \odot \Phi^*) \, \mathbf{F}^{(j)} - y_t - \epsilon_t + \eta_t^{(j)} \right)^2 \right] \\
&= \mathbb{E}_t \left[\left(\epsilon_t - \eta_t^{(j)} \right)^2 \right] \\
&= \mathbb{E}_t[\epsilon_t^2] + \mathbb{E}_t[\eta_t^{(j)2}] && (\mathbb{E}_t[\epsilon_t] = 0) \\
&= \sigma^2 + e \\
&= \widetilde{R}^{(j)}(\hat{\boldsymbol{W}}^{(j)}(\Phi^*), \Phi^*) + e \,,
\end{aligned}$$

for some $e > 0$. Then,

$$\frac{1}{M} \sum_{i=1}^{M} \widetilde{R}^{(i)}(\hat{\boldsymbol{W}}^{(i)}(\hat{\Phi}), \hat{\Phi}, \boldsymbol{\xi}) - \frac{1}{M} \sum_{i=1}^{M} \widetilde{R}^{(i)}(\hat{\boldsymbol{W}}^{(i)}(\Phi^*), \Phi^*, \boldsymbol{\xi}) > \frac{e}{M} \,, \tag{8}$$

and $||\Phi^*||_1 - ||\hat{\Phi}||_1 \leq m - 1$ (in the extreme case, when Φ^* contains all m basis functions, but $\hat{\Phi}$ uses a single basis function). Then choosing any $0 < \lambda_\Phi < \frac{e}{M(m-1)}$, we can rewrite Equation (8) as

$$\begin{aligned}
\frac{1}{M} \sum_{i=1}^{M} \widetilde{R}^{(i)}(\hat{\boldsymbol{W}}^{(i)}(\hat{\Phi}), \hat{\Phi}, \boldsymbol{\xi}) - \frac{1}{M} \sum_{i=1}^{M} \widetilde{R}^{(i)}(\hat{\boldsymbol{W}}^{(i)}(\Phi^*), \Phi^*, \boldsymbol{\xi}) &> \frac{e}{M} \\
&> \lambda_\Phi (m - 1) \\
&> \lambda_\Phi (||\Phi^*||_1 - ||\hat{\Phi}||_1) \,.
\end{aligned}$$

Rearranging the terms, we get $\mathcal{L}(\hat{\Phi}) > \mathcal{L}(\Phi^*)$, leading to contradiction.

In both cases above, we have shown there exists a small enough $\lambda_\Phi > 0$ such that the true causal structure Φ^* is the unique minimizer of $\mathcal{L}(\Phi)$.

□

B DESCRIPTION OF TASKS

For each dynamical system, we simulate the respective ODE to generate $M = 1000$ training tasks each observed over regularly-spaced discrete time steps $\{t_0, \ldots, t_T\}$ where $\forall l, t_l = 0.1l$. Our data generation process is succinctly depicted in Table 4. For each dataset, the second column shows the state variables $\mathbf{X}_t$ and the unknown parameters $\boldsymbol{W}^*$. For each training task $\mathcal{T}^{(i)}, i = 1, \ldots, M$, we sample an initial condition $\mathbf{X}_{t_0}^{(i)} \sim P^{(\text{tr})}(\mathbf{X}_{t_0})$ (shown under ID columns of the table). We sample a different $\boldsymbol{W}^{(i)*} \sim \mathcal{U}(\boldsymbol{W}_{\text{param}}, 2\boldsymbol{W}_{\text{param}})$ for each task i with $\boldsymbol{W}_{\text{param}}$ shown in Table 4.

At test, we generate $M' = 200$ test tasks by simulating the respective dynamical system over timesteps $\{t_0, \ldots, t_r\}$, where again $\forall l, t_l = 0.1l$. For each test task $j = 1, \ldots, M'$, we sample initial conditions $\mathbf{X}_{t_0}^{(j)} \sim P^{(\text{te})}(\mathbf{X}_{t_0})$ with a completely different support for the initial conditions $\mathbf{X}_{t_0}^{(j)}$ than in training. The distribution of the dynamical system parameters $\boldsymbol{W}^*$ is kept the same for "OOD $\mathbf{X}_{t_0}$" scenario but is shifted for "OOD $\mathbf{X}_{t_0}$ and $\boldsymbol{W}^*$" scenario. In the latter, we sample a different $\boldsymbol{W}^{(j)*} \sim \mathcal{U}(2\boldsymbol{W}_{\text{param}}, 3\boldsymbol{W}_{\text{param}})$ for each test task j with $\boldsymbol{W}_{\text{param}}$ shown in Table 4.

Damped pendulum system (Yin et al., 2021). The state $\mathbf{X}_t = [\theta_t, \omega_t] \in \mathbb{R}^2$ describes the angle made by the pendulum with the vertical and the corresponding angular velocity at time t. The true (unknown) function ψ describing this dynamical system is given by $\frac{d\theta_t}{dt} = \omega_t$, $\frac{d\omega_t}{dt} = -\alpha^{*2} \sin(\theta_t) - \rho^* \omega_t$ where $\boldsymbol{W}^* = (\alpha^*, \rho^*)$ are the dynamical system parameters. We simulate the ODE over time steps $\{t_0, \ldots, t_T\}$ with $\forall l, t_l = 0.1l, T = 100$ in training and over time steps $\{t_0, \ldots, t_r\}$ in test with $r = \frac{1}{3}T$. In training, the pendulum is dropped from initial angles $\theta_{t_0}^{(i)} \sim \mathcal{U}(0, \pi/2)$ with no angular velocity, whereas in OOD test, the pendulum is dropped from initial angles $\theta_{t_0}^{(j)} \sim \mathcal{U}(\pi - 0.1, \pi)$ and angular velocity $\omega_{t_0}^{(j)} \in \mathcal{U}(-1, 0)$.

Predator-prey system (Wang et al., 2021a). We wish to model the dynamics between two species acting as prey and predator respectively. We adapt the experiment by Wang et al. (2021a) to our out-of-distribution forecasting scenario according to Definition 1. Let p and q denote the prey and predator populations respectively. The ordinary differential equations describing the dynamical system is given by $\frac{dp}{dt} = \alpha^* p - \beta^* pq$, $\frac{dq}{dt} = \delta^* pq - \gamma^* q$, where $\boldsymbol{W}^* = (\alpha^*, \beta^*, \gamma^*, \delta^*)$ are the (unknown) dynamical system parameters. We simulate the ODE over time steps $\{t_0, \ldots, t_T\}$ with $\forall l, t_l = 0.1l, T = 100$ in training and over time steps $\{t_0, \ldots, t_r\}$ in test with $r = \frac{1}{3}T$. We generate $M = 1000$ training tasks with different initial prey and predator populations with prey $p_{t_0}^{(i)} \sim \mathcal{U}(1000, 2000)$ and predator $q_{t_0}^{(i)} \sim \mathcal{U}(10, 20)$ for each $i = 1, \ldots, M$. At OOD test, we generate $M' = 200$ out-of-distribution (OOD) test tasks with different initial prey populations $p_{t_0}^{(j)} \sim \mathcal{U}(100, 200)$ but the same distribution for predator population $q_{t_0}^{(j)} \sim \mathcal{U}(10, 20)$.

Epidemic modeling (Wang et al., 2021a). We adapt the experiment by Wang et al. (2021a) to our out-of-distribution forecasting scenario according to Definition 1. The state of the dynamical system is described by three variables: number of susceptible (S), infected (I) and recovered (R) individuals. The dynamics is described using the following ODEs: $\frac{dS}{dt} = -\beta \frac{SI}{N}$, $\frac{dI}{dt} = \beta \frac{SI}{N} - \gamma I$, $\frac{dR}{dt} = \gamma I$, where $\boldsymbol{W} = (\beta, \gamma)$ are the (unknown) dynamical system parameters and $N = S + I + R$ is the total population. We simulate the ODE over time steps $\{t_0, \ldots, t_T\}$ with $\forall l, t_l = 0.1l, T = 100$ in training and over time steps $r = \frac{1}{10}T$. We generate $M = 1000$ training tasks with different initial populations for susceptible (S) and infected (I) individuals, while the number of initial recovered (R) individuals are always zero. In training, we sample $S_{t_0}^{(i)} \sim \mathcal{U}(9, 10)$ and $I_{t_0}^{(i)} \sim \mathcal{U}(1, 5)$ for each $i = 1, \ldots, M$. At OOD test, we generate $M' = 200$ out-of-distribution test tasks with a different initial susceptible population, $S_{t_0}^{(j)} \sim \mathcal{U}(90, 100)$, while keeping the same distribution for infected population.

C IMPLEMENTATION DETAILS

In what follows, we describe implementation details of MetaPhysiCa and the baselines.

Table 2: **(Damped pendulum results)** Normalized RMSE ↓ of test predictions from different methods in-distribution and two OOD scenarios. NaN* indicates that the model returned errors during test-time predictions, for example, because the learnt ODE was too stiff (numerically unstable) to solve.

Methods	Test NRMSE ↓ ID	OOD $\mathbf{X}_{t_0}$	OOD $\mathbf{X}_{t_0}$ and $\boldsymbol{W}^*$
Standard Deep Learning			
NeuralODE (Chen et al., 2018)	0.083 (0.033)	0.591 (0.119)	0.717 (0.210)
Meta Learning			
DyAd (Wang et al., 2021b)	0.078 (0.051)	0.834 (0.263)	0.804 (0.267)
CoDA (Kirchmeyer et al., 2022)	0.052 (0.032)	0.764 (0.201)	1.011 (0.226)
Physics-informed Machine Learning			
APHYNITY (Yin et al., 2021)	0.097 (0.020)	0.970 (0.384)	1.159 (0.334)
SINDy (Brunton et al., 2016)	NaN*	NaN*	NaN*
EQL (Martius & Lampert, 2016)	NaN*	NaN*	NaN*
MetaPhysiCa **(ours)**	0.049 (0.002)	**0.070 (0.011)**	**0.181 (0.012)**

Table 3: **(Predator-prey results)** Test NRMSE ↓ for different methods. NaN* indicates that the model returned errors during test. Standard deep learning methods and physics-informed deep learning methods fail to forecast accurately out-of-distribution. **MetaPhysiCa outputs** $8\times$ **and** $2\times$ **more robust OOD predictions** in the two OOD scenarios respectively.

Methods	Test Normalized RMSE (NRMSE) ↓ ID	OOD $\mathbf{X}_{t_0}$	OOD $\mathbf{X}_{t_0}$ and $\boldsymbol{W}^*$
Standard Deep Learning			
NeuralODE (Chen et al., 2018)	0.193 (0.024)	1.056 (0.141)	0.969 (0.172)
Meta Learning			
DyAd (Wang et al., 2021b)	0.244 (0.025)	1.088 (0.373)	1.025 (0.403)
CoDA (Kirchmeyer et al., 2022)	NaN*	NaN*	NaN*
Physics-informed Machine Learning			
APHYNITY (Yin et al., 2021)	0.421 (0.332)	3.937 (1.686)	1.281 (0.457)
SINDy (Brunton et al., 2016)	NaN*	NaN*	NaN*
EQL (Martius & Lampert, 2016)	NaN*	NaN*	NaN*
MetaPhysiCa **(Ours)**	0.049 (0.008)	**0.129 (0.030)**	**0.434 (0.128)**

Datasets	State variables	ID	OOD $\mathbf{X}_{t_0}$	OOD $\mathbf{X}_{t_0}$ and $\boldsymbol{W}^*$
Damped pendulum	$\mathbf{X}_t = (\theta_t, \omega_t)$ $\boldsymbol{W}^* = (\alpha, \rho)$	$\theta_0 \sim \mathcal{U}(0, \pi/2)$ $\omega_0 = 0$	$\theta_0 \sim \mathcal{U}(\pi - 0.1, \pi)$ $\omega_0 \sim \mathcal{U}(-1, 0)$ $\alpha_{\text{param}} = 1, \rho_{\text{param}} = 0.2$	$\theta_0 \sim \mathcal{U}(\pi - 0.1, \pi)$ $\omega_0 \sim \mathcal{U}(-1, 0)$
Predator prey system	$\mathbf{X}_t = (p_t, q_t)$ $\boldsymbol{W}^* = (\alpha, \beta, \gamma, \delta)$	$p_0 \sim \mathcal{U}(1000, 2000)$ $q_0 \sim \mathcal{U}(10, 20)$	$p_0 \sim \mathcal{U}(100, 200)$ $q_0 \sim \mathcal{U}(10, 20)$ $\alpha_{\text{param}} = 1, \beta_{\text{param}} = 0.06, \gamma_{\text{param}} = 0.5, \delta_{\text{param}} = 0.0005$	$p_0 \sim \mathcal{U}(100, 200)$ $q_0 \sim \mathcal{U}(10, 20)$
Epidemic modeling	$\mathbf{X}_t = (S_t, I_t, R_t)$ $\boldsymbol{W}^* = (\beta, \gamma)$	$S_0 \sim \mathcal{U}(9, 10)$ $I_0 \sim \mathcal{U}(1, 5)$ $R_0 = 0$	$S_0 \sim \mathcal{U}(90, 100)$ $I_0 \sim \mathcal{U}(1, 5)$ $R_0 = 0$ $\beta_{\text{param}} = 4, \gamma_{\text{param}} = 0.4$	$S_0 \sim \mathcal{U}(90, 100)$ $I_0 \sim \mathcal{U}(1, 5)$ $R_0 = 0$

Table 4: Description of the dataset generation process. For each dataset, $\mathbf{X}_t$ denotes the state variable of the dynamical system and $\boldsymbol{W}^*$ denotes its parameters. Column "ID" represents in-distribution initial states while the last two columns represent the two out-of-distribution scenarios. In-distribution ODE parameters $\boldsymbol{W}^{(i)*}$ are sampled from a uniform distribution $\boldsymbol{W}^{(i)*} \sim \mathcal{U}(\boldsymbol{W}_{\text{param}}, 2\boldsymbol{W}_{\text{param}})$ and the out-of-distribution ODE parameters are sampled as $\boldsymbol{W}^{(i)*} \sim \mathcal{U}(2\boldsymbol{W}_{\text{param}}, 3\boldsymbol{W}_{\text{param}})$. For example, in the damped pendulum dataset, in-distribution parameters are sampled as $\alpha^{(i)*} \sim \mathcal{U}(\alpha_{\text{param}}, 2\alpha_{\text{param}}) = (1, 2)$ and $\rho^{(i)*} \sim \mathcal{U}(\rho_{\text{param}}, 2\rho_{\text{param}}) = (0.2, 0.4)$ for each task i. Similarly, the out-of-distribution ODE parameters (in the last column) are sampled as $\alpha^{(i)*} \sim \mathcal{U}(2\alpha_{\text{param}}, 3\alpha_{\text{param}}) = (2, 3)$ and $\rho^{(i)*} \sim \mathcal{U}(2\rho_{\text{param}}, 3\rho_{\text{param}}) = (0.4, 0.6)$.

C.1 METAPHYSICA

Recall from Equation (3) that the proposed model is defined as

$$\frac{d\hat{\mathbf{X}}_t^{(i)}}{dt} = (\boldsymbol{W}^{(i)} \odot \Phi) F(\hat{\mathbf{X}}_t^{(i)}; \boldsymbol{\xi}) \,, \tag{9}$$

where $\odot$ is the Hadamard product and

- $F(\hat{\mathbf{X}}_t^{(i)}; \boldsymbol{\xi}) := \begin{bmatrix} f_1(\hat{\mathbf{X}}_t^{(i)}; \boldsymbol{\xi}_1) & \cdots & f_m(\hat{\mathbf{X}}_t^{(i)}; \boldsymbol{\xi}_m) \end{bmatrix}^T$ is the vector of outputs from the basis functions with parameters $\boldsymbol{\xi}$,
- $\Phi \in \{0, 1\}^{d \times m}$ are the learnable parameters governing the global causal structure across all tasks such that $\Phi_{j,k} = 1$ iff edge $z_{k,t} \to {}^{d\boldsymbol{x}_{t,j}}/_{dt}$ exists,
- $\boldsymbol{W}^{(i)} \in \mathbb{R}^{d \times m}$ are task-specific parameters that act as coefficients in linear combination of the selected basis functions.

In our experiments, we use polynomial and trigonometric basis functions, such that

$$F(\hat{\mathbf{X}}_t^{(i)}; \boldsymbol{\xi}) := \left[1 \quad \underbrace{\hat{\mathbf{X}}_{t,1}^{(i)} \dots \hat{\mathbf{X}}_{t,d}^{(i)}}_{\text{polynomial order 1}} \quad \underbrace{\hat{\mathbf{X}}_{t,1}^{(i)2} \dots \hat{\mathbf{X}}_{t,l-1}^{(i)} \hat{\mathbf{X}}_{t,l}^{(i)} \dots \hat{\mathbf{X}}_{t,d}^{(i)2}}_{\text{polynomial order 2}} \quad \underbrace{\sin(\xi_{1,1}\hat{\mathbf{X}}_{t,1}^{(i)} + \xi_{1,2}) \dots \sin(\xi_{d,1}\hat{\mathbf{X}}_{t,d}^{(i)} + \xi_{d,2})}_{\text{trigonometric}} \right]^T .$$

Equation (4) describes a bi-level objective that optimizes the structure parameters Φ and the global parameters ξ in the outer-level, and the task-specific parameters $\boldsymbol{W}^{(i)}$ in the inner-level as follows

$$\hat{\Phi}, \hat{\boldsymbol{\xi}} = \arg\min_{\Phi, \boldsymbol{\xi}} \frac{1}{M} \sum_{i=1}^{M} R^{(i)}(\hat{\boldsymbol{W}}^{(i)}, \Phi, \boldsymbol{\xi}) + \lambda_\Phi ||\Phi||_1 + \lambda_{\text{REx}} \text{Variance}(\{R^{(i)}(\hat{\boldsymbol{W}}^{(i)}, \Phi, \boldsymbol{\xi})\}_{i=1}^{M})$$
$$\text{s.t. } \hat{\boldsymbol{W}}^{(i)} = \arg\min_{\boldsymbol{W}^{(i)}} R^{(i)}(\boldsymbol{W}^{(i)}, \Phi, \boldsymbol{\xi}) \quad \forall i = 1, \dots, M \,,$$

where λ_Φ and λ_{REx} are hyperparameters. As discussed in the main text, the jointly optimizing $\Phi, \boldsymbol{\xi}$ and $\boldsymbol{W}^{(i)}, i = 1, \ldots, M$, instead of alternating SGD resulted in comparable performance with considerable computational benefits. We use the following joint optimization objective to approximate Equation (4),

$$\hat{\Phi}, \hat{\boldsymbol{\xi}}, \hat{\boldsymbol{W}}^{(1)}, \ldots, \hat{\boldsymbol{W}}^{(M)} = \underset{\Phi, \boldsymbol{\xi}, \boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(M)}}{\arg\min} \frac{1}{M} \sum_{i=1}^{M} R^{(i)}(\boldsymbol{W}^{(i)}, \Phi, \boldsymbol{\xi}) + \lambda_\Phi ||\Phi||_1 + \lambda_{\text{REx}} \text{Variance}(\{R^{(i)}(\boldsymbol{W}^{(i)}, \Phi, \boldsymbol{\xi})\}_{i=1}^{M}) \tag{10}$$

We perform a grid search over the following hyperparameters: regularization strengths $\lambda_\Phi \in \{10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}\}, \lambda_{\text{REx}} \in \{0, 10^{-2}, 10^{-1}, 1, 10\}$, and learning rates $\eta \in \{10^{-2}, 10^{-3}, 10^{-4}\}$. We use a linearly increasing scheduler for λ_{REx} to avoid the V-REx penalty dominating the loss during initial epochs. We choose the hyperparameters that result in sparsest model (i.e., with the least $||\hat{\Phi}||_0$) while achieving validation loss within 5% of the best validation loss in held-out *in-distribution* validation data.

Time complexity. For a single task i and time t, computing the predicted derivatives $d\hat{X}_t^{(i)}/dt$ in Equation (3) has $O(md)$ complexity to evaluate the Hadamard product and the matrix-vector product, where m is the number of basis functions and d is the state dimension of the dynamical system. Since there are M tasks/trajectories of maximum length T, the overall complexity per epoch is $O(mdMT)$. However MetaPhysiCa could require a higher number of epochs to converge for higher values of m (Figure 9).

C.2 NEURALODE (CHEN ET AL., 2018)

The prediction dynamics corresponding to the latent NeuralODE model is given by $\frac{d\hat{\mathbf{X}}_t}{dt} = F_{\text{nn}}(\hat{\mathbf{X}}_t, \boldsymbol{z}_{\leq r}; \boldsymbol{W}_1)$ where $\boldsymbol{z}_{\leq r} = F_{\text{enc}}(\mathbf{X}_{t_0}, \ldots, \mathbf{X}_{t_r}; \boldsymbol{W}_2)$ encodes the initial observations using a recurrent neural network F_{enc} (e.g., GRU), and F_{nn} is a feedforward neural network. The model is trained with an ODE solver (dopri5) and the gradients computed using the adjoint method (Chen et al., 2018). We perform a grid search over the following hyperparameters: number of layers for F_{nn}, $L \in \{1, 2, 3\}$, size of each hidden layer of F_{nn}, $d_h \in \{32, 64, 128\}$, size of the encoder representation $\boldsymbol{z}_{\leq r}$, $d_z \in \{32, 64, 128\}$, batch sizes $B \in \{32, 64\}$, and learning rates $\eta \in \{10^{-2}, 10^{-3}, 10^{-4}\}$.

C.3 DYAD (MODIFIED FOR ODES) (WANG ET AL., 2021B)

DyAd, originally proposed for forecasting PDEs, uses a meta-learning framework to adapt to different training tasks by learning a per-task weak label. We modify their approach for our ODE-based experiments. Since we do not assume the presence of weak labels for supervision for adaptation, we use mean of each variable in the training task as the task's weak label. We use NeuralODE as the base sequence model for the forecaster network. The forecaster network takes the initial observations as input and forecasts the future observations while being adapted with the encoder network. The encoder network is a recurrent network (GRU in our experiments) that takes as input the initial observations and predicts the weak label. The last layer representation from the encoder network is used to adapt NeuralODE via AdaIN (Huang & Belongie, 2017). We perform a grid search over the following hyperparameters: size of hidden layers for the forecaster and encoder networks $d_h \in \{32, 64, 128\}$, number of layers for the forecaster network, $L \in \{1, 2, 3\}$, batch sizes $B \in \{32, 64\}$, and learning rates $\eta \in \{10^{-2}, 10^{-3}, 10^{-4}\}$.

C.4 APHYNITY (YIN ET AL., 2021)

APHYNITY assumes that we are given a (possibly incomplete) physics model $\phi(\cdot, \Theta_{\text{phy}})$ with parameters Θ_{phy}. When the training data may consist of tasks with different $\boldsymbol{W}^{(i)}*$, APHYNITY predicts the physics parameters with respect to the task i inductively using a recurrent neural network G_{nn} from the initial observations of the system as $\hat{\Theta}_{\text{phy}}^{(i)} = G_{\text{nn}}(\mathbf{X}_{t_0}, \ldots, \mathbf{X}_{t_r}; \boldsymbol{W}_2)$. Then, APHYNITY augments the given physics model ϕ with a feedforward neural network component F_{nn} and defines the final dynamics as $\frac{d\hat{\mathbf{X}}_t^{(i)}}{dt} = \phi(\hat{\mathbf{X}}_t^{(i)}; \hat{\Theta}_{\text{phy}}^{(i)}) + F_{\text{nn}}(\hat{\mathbf{X}}_t^{(i)}; \boldsymbol{W}_1)$. APHYNITY solves a

constrained optimization problem to minimize the norm of the neural network component while still predicting the training trajectories accurately. The model is trained with an ODE solver (dopri5) and the gradients computed using the adjoint method (Chen et al., 2018). In our experiments, we provide APHYNITY with simpler physics models:

- For damped pendulum system, we use a physics model that assumes no friction: $\frac{d\theta_t}{dt} = \omega_t$, $\frac{d\omega_t}{dt} = -\alpha^2_{\text{phy}} \sin(\theta_t)$ where $\Theta_{\text{phy}} = \alpha_{\text{phy}}$ is the physics model parameter.
- For predator-prey system, we use a physics model that assumes no interaction between the two species: $\frac{dp}{dt} = \alpha_{\text{phy}} p$, $\frac{dq}{dt} = -\gamma_{\text{phy}} q$ where $\Theta_{\text{phy}} = (\alpha_{\text{phy}}, \gamma_{\text{phy}})$ are the physics model parameters.
- For epidemic model, we use a physics model that assumes the disease is not infectious: $\frac{dS}{dt} = 0$, $\frac{dI}{dt} = -\gamma I$, $\frac{dR}{dt} = \gamma I$, where $\Theta_{\text{phy}} = \gamma_{\text{phy}}$ is the physics model parameter.

In each dataset, APHYNITY needs to augment the physics model with a neural network component for accurate predictions.

We perform a grid search over the following hyperparameters: number of layers for F_{nn}, $L \in \{1, 2, 3\}$, size of each hidden layer of F_{nn}, $d_h \in \{32, 64, 128\}$, batch sizes $B \in \{32, 64\}$, and learning rates $\eta \in \{10^{-2}, 10^{-3}, 10^{-4}\}$.

C.5 SINDy (Brunton et al., 2016)

SINDy uses a given dictionary of basis functions to model the dynamics as $\frac{d\hat{\mathbf{X}}_t}{dt} = \Theta(\hat{\mathbf{X}}_t)\boldsymbol{W}$ where Θ is feature map with the basis functions (such as polynomial and trigonometric functions) and $\boldsymbol{W}$ is simply a weight matrix. SINDy is trained using sequential threshold least squares (STLS) for sparse weights $\boldsymbol{W}$. We perform a grid search over the following hyperparameters: threshold parameter used in STLS optimization, $\tau_0 \in \{0.005, 0.01, 0.05, 0.1, 0.2, 0.5\}$, and the regularization strength $\alpha \in \{0.05, 0.01, 0.1, 0.5\}$.

C.6 Equation Learner (Martius & Lampert, 2016)

Equation learner (EQL) is a neural network architecture where each layer is defined as follows with input $\boldsymbol{x}$ and output $\boldsymbol{o}$

$$\begin{aligned} \boldsymbol{z} &= \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b} \\ \boldsymbol{o} &= (f_1(z_1), f_2(z_2), \dots, g_1(z_k, z_{k+1}), g_2(z_{k+2}, z_{k+3}), \dots,) \,, \end{aligned}$$

where f_i are unary basis functions (such as $\sin$, $\cos$, etc.) and g_i are binary basis functions (such as multiplication). We use $\text{id}, \sin$ and multiplication functions in our implementation. EQL is trained using a sparsity inducing ℓ_1-regularization with hard thresholding for the final few epochs. We perform a grid search over the following hyperparameters: number of EQL layers, $L \in \{1, 2\}$, number of nodes for each type of basis function, $h \in \{1, 3, 5\}$, regularization strength $\alpha \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$, batch sizes $B \in \{32, 64\}$, and learning rates $\eta \in \{10^{-2}, 10^{-3}, 10^{-4}\}$.

D Additional results

D.1 Qualitative analysis

Recall from Equation (3) that the proposed model is defined as

$$\frac{d\hat{\mathbf{X}}^{(i)}_t}{dt} = (\boldsymbol{W}^{(i)} \odot \Phi) F(\hat{\mathbf{X}}^{(i)}_t; \boldsymbol{\xi}) \,, \tag{11}$$

where $F(\hat{\mathbf{X}}^{(i)}_t; \boldsymbol{\xi})$ is the vector of outputs from the basis functions, $\Phi \in \{0, 1\}^{d \times m}$ are the learnable parameters governing the global causal structure across all tasks, and $\boldsymbol{W}^{(i)} \in \mathbb{R}^{d \times m}$ are task-specific parameters that act as coefficients in linear combination of the selected basis functions.

Datasets	State variables	Ground truth ODE	Learnt ODE (from Φ)
Damped pendulum	$\mathbf{X}_t = (\theta_t, \omega_t)$	$\frac{d\theta_t}{dt} = \omega_t$ $\frac{d\omega_t}{dt} = -\alpha^{*2}\sin(\theta_t) - \rho^*\omega_t$	$\frac{d\theta_t}{dt} = W_1\omega_t$ $\frac{d\omega_t}{dt} = W_2\sin(\theta_t) + W_3\omega_t$
Predator prey system	$\mathbf{X}_t = (p_t, q_t)$	$\frac{dp_t}{dt} = \alpha^* p_t - \beta^* p_t q_t$ $\frac{dq_t}{dt} = \delta^* p_t q_t - \gamma^* q_t$	$\frac{dp_t}{dt} = W_1 p_t + W_2 p_t q_t$ $\frac{dq_t}{dt} = W_3 p_t q_t + W_4 q_t$
Epidemic modeling	$\mathbf{X}_t = (S_t, I_t, R_t)$	$\frac{dS_t}{dt} = -\beta^*\frac{S_t I_t}{S_t+I_t+R_t}$ $\frac{dI_t}{dt} = \beta^*\frac{S_t I_t}{S_t+I_t+R_t} - \gamma^* I_t$ $\frac{dR_t}{dt} = \gamma^* I_t$	$\frac{dS_t}{dt} = W_1 S_t I_t$ $\frac{dI_t}{dt} = W_2 S_t I_t + W_3 I_t^2 + W_4 I_t R_t$ $\frac{dR_t}{dt} = W_5 S_t I_t + W_6 I_t^2 + W_7 I_t R_t$

Table 5: **(Qualitative analysis.)** Ground truth dynamical system vs learnt ODE in the meta-model Φ. Recall that $\Phi \in \{0,1\}^{d\times m}$ dictates which of the basis functions affect the output ${}^{d\mathbf{X}_t}/_{dt}$. The weights W_l in the learnt ODE column are learnable parameters that are optimized via test-time adaptation in Equation (5). **MetaPhysiCa learns the exact ground truth ODE for Damped pendulum and Predator-prey system, and a reparameterized version of the true ODE for epidemic modeling task.**

After training, the ODE learnt by the model can be easily inferred by checking all the terms in Φ that are greater than zero, i.e., $\Phi_{j,k} > 0$ implies $f_k(\boldsymbol{x}_t; \boldsymbol{\xi}_k) \rightarrow {}^{d\boldsymbol{x}_{t,j}}/_{dt}$ exists in the causal graph. In other words, RHS of learnt ODE for ${}^{d\boldsymbol{x}_{t,j}}/_{dt}$ contains the basis function $f_k(\boldsymbol{x}_t; \boldsymbol{\xi}_k)$.

Table 5 shows the ground truth ODE and the learnt ODE for the three experiments. For each learnt ODE, we also depict the learnable parameters W_l that can be adapted using Equation (5) during test-time. For damped pendulum and predator-prey system, the RHS terms in the learnt ODE exactly matches ground truth ODE, and from Figures 3 and 7, it is clear that the method is able to accurately adapt the learnable parameters W_l during test-time. For epidemic modeling task, MetaPhysiCa learns a reparameterized version of the ground truth ODE. For example, MetaPhysiCa learns $\frac{dR_t}{dt} = W'_a I_t S_t + W'_b I_t^2 + W'_c I_t R_t$, which can be written as $\frac{dR_t}{dt} = W_a I_t$ (the ground truth ODE) if $W'_a = W'_b = W'_c$, because $S_t + I_t + R_t = N$ is a constant denoting the total population. While the learnt reparameterized ODE is more complex because it allows different values for W'_a, W'_b, W'_c, the test-time adaptation of these learnable parameters with the initial test observations results in them taking the same values.

D.2 Ablation results

D.2.1 All components

We present an ablation study comparing different components of MetaPhysiCa in Table 6. Table shows out-of-distribution test NRMSE for MetaPhysiCa without each individual component on the three dynamical systems (OOD w.r.t $\mathbf{X}_{t_0}$). We observe that sparsity regularization (i.e., $||\Phi||_1$) and test-time adaptation are the most important components. For two out of three tasks, the method returns prediction errors without sparsity regularization.

When testing MetaPhysiCa without test-time adaptation, we simply use the mean of the task-specific weights learnt for training tasks as the task-specific weight for the given test trajectory, i.e., $\hat{\boldsymbol{W}}^{M+1} = \frac{1}{M}\sum_i \boldsymbol{W}^{(i)}$. This results in high OOD errors showing the importance of test-time adaptation. V-REx penalty (Krueger et al., 2021) helps in some experiments and performs comparably in others.

D.2.2 V-REx penalty

We repeat the damped pendulum experiment described in Appendix B with modified training distributions. Recall that the true function ψ describing this dynamical system is given by $\frac{d\theta_t}{dt} = \omega_t$, $\frac{d\omega_t}{dt} = -\alpha^{*2}\sin(\theta_t) - \rho^*\omega_t$ where ρ^* is the damping coefficient. In training, the pendulum is dropped from initial angles $\theta_{t_0}^{(i)} \sim \mathcal{U}(0, \pi/2)$ with no angular velocity, whereas in OOD test, the pendulum is dropped from initial angles $\theta_{t_0}^{(j)} \sim \mathcal{U}(\pi - 0.1, \pi)$.

Method	Test Normalized RMSE ↓ (OOD $\mathbf{X}_{t_0}$) Damped Pendulum	Predator-Prey	Epidemic Modeling
MetaPhysiCa	**0.070 (0.011)**	**0.129 (0.030)**	**0.019 (0.002)**
without $\|\|\Phi\|\|_1$	NaN*	1.806 (0.736)	NaN*
without test-time adaptation	1.223 (0.741)	1.404 (3.794)	0.358 (0.554)
without V-REx penalty	**0.070 (0.014)**	**0.129 (0.030)**	0.042 (0.065)

Table 6: **(Ablation.)** Out-of-distribution test NRMSE for MetaPhysiCa without each individual component on the three dynamical systems (OOD w.r.t. $\mathbf{X}_{t_0}$ alone). **Sparsity regularization (i.e., $||\Phi||_1$) and test-time adaptation are the most important components.**

Table 7: **(Damped pendulum.)** True casual model for the damped pendulum is given by $d\theta_t/dt = \omega_t;\ d\omega_t/dt = -\alpha^{*2}\sin(\theta_t) - \rho^*\omega_t$, where ρ^* is the damping coefficient. In training, we have $P^{(\text{tr})}(\rho^* > 0) = 10^{-2}$, i.e., most training tasks have no damping. In OOD test, with arbitrary interventions allowed on ρ^*, we consider $P^{(\text{te})}(\rho^* > 0) = 0.5$. **MetaPhysiCa without V-REx penalty sacrifices performance on 1% of training tasks and learns the incorrect ODE with no damping. With V-Rex penalty, the risks of all tasks are forced to be close to equal, and the model learns the true causal structure.**

Method	Learnt $\hat{\Phi}$	Test Normalized RMSE ↓ ID	OOD $\mathbf{X}_{t_0}$ and $\boldsymbol{W}^*$
MetaPhysiCa			
without V-REx penalty	$\frac{d\theta_t}{dt} = W_1\omega_t$ $\frac{d\omega_t}{dt} = W_2\sin(\theta_t)$	**0.053 (0.009)**	0.829 (0.106)
with V-REx penalty	$\frac{d\theta_t}{dt} = W_1\omega_t$ $\frac{d\omega_t}{dt} = W_2\sin(\theta_t) + W_3\omega_t$	**0.049 (0.008)**	**0.034 (0.007)**

In training, we sample the damping coefficient ρ^* from a mixture distribution $P^{(\text{tr})}$ such that $\rho^* = 0$ with probability 0.99 and $\rho^* \sim \mathcal{U}(0.2, 0.4)$ with probability 0.01. This means that very few training tasks have damping, but the model should still learn the true causal structure with the damping term. In OOD test, with arbitrary interventions allowed on ρ^*, we consider $P^{(\text{te})}(\rho^* > 0) = 0.5$.

Table 7 shows that MetaPhysiCa without V-REx penalty sacrifices performance on 1% of training tasks and learns the incorrect ODE with no damping. On the other hand, MetaPhysiCa with V-REx penalty forces the risks of all tasks to be close to equal (including the 1% tasks with damping), and thus learns the true causal structure.

D.2.3 Bi-level optimization vs Joint optimization

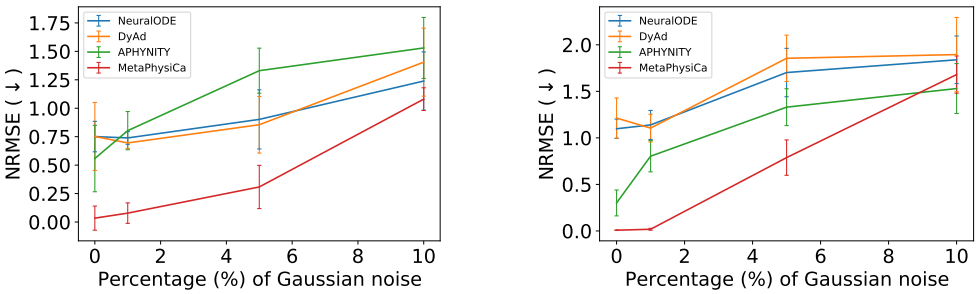
Recall that Equation (4) describes a bi-level objective that optimizes the structure parameters Φ and the global parameters ξ in the outer-level, and the task-specific parameters $\boldsymbol{W}^{(i)}$ in the inner-level as follows

$$\hat{\Phi}, \hat{\boldsymbol{\xi}} = \arg\min_{\Phi, \boldsymbol{\xi}} \frac{1}{M}\sum_{i=1}^{M} R^{(i)}(\hat{\boldsymbol{W}}^{(i)}, \Phi, \boldsymbol{\xi}) + \lambda_\Phi ||\Phi||_1 + \lambda_{\text{REx}}\text{Variance}(\{R^{(i)}(\hat{\boldsymbol{W}}^{(i)}, \Phi, \boldsymbol{\xi})\}_{i=1}^{M})$$
$$\text{s.t. } \hat{\boldsymbol{W}}^{(i)} = \arg\min_{\boldsymbol{W}^{(i)}} R^{(i)}(\boldsymbol{W}^{(i)}, \Phi, \boldsymbol{\xi}) \quad \forall i = 1, \ldots, M\,,$$

where λ_Φ and λ_{REx} are hyperparameters.

We evaluate two ways to optimize the above: (a) bi-level optimization using inner and outer loops, and (b) joint optimization of all variables together.

Bi-level optimization. We use two loops where the outer loop optimizes the global parameters $\hat{\Phi}$ and $\hat{\xi}$, and the inner loop optimizes the task-specific parameters $\boldsymbol{W}^{(i)*}$ for all i while keeping



(a) Damped pendulum task

(b) Predator-prey system

Figure 8: **(Performance with increasing noise.)** Out-of-distribution NRMSE values for Damped Pendulum and Predator-prey experiments with different percentages of Gaussian noise added $(0\%, 1\%, 5\%, 10\%)$. **MetaPhysiCa is relatively robust to $\leq 5\%$ Gaussian noise and outperforms the baselines. With a larger amount of noise, MetaPhysiCa is unable to identify the dynamical system accurately but performs comparable to the baselines.**

the global parameters fixed. Rather than running the inner loop till convergence, we only perform a few gradient steps S of the inner optimization ($S = 10$ in our experiments). This is a standard meta-learning technique to improve computation complexity.

Joint optimization. We use the following joint optimization objective to approximate Equation (4),

$$\hat{\Phi}, \hat{\boldsymbol{\xi}}, \hat{\boldsymbol{W}}^{(1)}, \ldots, \hat{\boldsymbol{W}}^{(M)} = \underset{\Phi, \boldsymbol{\xi}, \boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(M)}}{\arg\min} \frac{1}{M} \sum_{i=1}^{M} R^{(i)}(\boldsymbol{W}^{(i)}, \Phi, \boldsymbol{\xi}) + \lambda_{\Phi} ||\Phi||_1 + \lambda_{\text{REx}} \text{Variance}(\{R^{(i)}(\boldsymbol{W}^{(i)}, \Phi, \boldsymbol{\xi})\}_{i=1}^{M}) \tag{12}$$

Both bi-level optimization with $S = 10$ and joint optimization in Equation (12) result in $\hat{\Phi}$ learning the true dynamics for all 3 tested dynamical systems. However, joint optimization is $8.2\times$ faster per epoch than the bi-level optimization (taking 90ms vs 744ms on average on one Intel(R) Xeon(R) CPU core).

D.3 Robustness to noise

We repeat the Damped pendulum and Predator-prey experiments with increasing amounts of noises. Specifically, we add $1\%, 5\%$ and 10% Gaussian noise to all the trajectories, both in training and in test. We use Total Variation Regularization (TVR) (Rudin et al., 1992; Chartrand, 2011) for estimating derivatives from noisy data as done by Brunton et al. (2016). We report the normalized RMSE for different models trained on the noisy versions of data in Figure 8. SINDy and EQL are not shown as they returned errors during test-time predictions similar to the case with no noise because the learnt ODE was too stiff (numerically unstable) to solve. In both tasks, the proposed method is relatively robust to small amounts of noise and outperforms the baselines. With 10% noise, MetaPhysiCa is unable to identify the dynamical system accurately, but performs comparable to the baselines.

D.4 Complex ODE Task

In this section, we extend MetaPhysiCa to consider significantly more expressive structural causal models (compared to Figure 4) that allow for composition of the basis functions. This is achieved with a 2-layer learnable basis function composition procedure. For example, given basis functions $f_1(\boldsymbol{x}_t; \xi_1) = \sin(\xi_{1,1}\boldsymbol{x}_{t,1} + \xi_{1,2})$, and $f_2(\boldsymbol{x}_t; \xi_2) = \boldsymbol{x}_{t,1}\boldsymbol{x}_{t,2}$, one can construct more expressive basis functions with compositions: $\tilde{f}_3(\boldsymbol{x}_t; \xi_3) = \sin(\xi_{3,3}\sin(\xi_{3,1}\mathbf{x}_{t,1} + \xi_{3,2}) + \xi_{3,4})$, $\tilde{f}_4(\boldsymbol{x}_t; \xi_4) = \boldsymbol{x}_{t,1}\boldsymbol{x}_{t,2}\sin(\xi_{4,1}\boldsymbol{x}_{t,1} + \xi_{4,2})$, etc., where ξ_j are global parameters that remain constant for all training/test tasks. The rest of the SCM remains the same and the derivative $d\mathbf{x}_{t,j}^{(i)}/dt$ for a particular dimension $j \in \{1, \ldots, d\}$ is a sparse linear combination of the original basis functions and the more expressive second layer ones.

Methods	Test Normalized RMSE (NRMSE) ↓ ID	OOD $\mathbf{X}_{t_0}$
NeuralODE (Chen et al., 2018)	0.034 (0.008)	0.296 (0.064)
APHYNITY (Yin et al., 2021)	0.027 (0.010)	0.684 (0.117)
MetaPhysiCa (Ours)	0.188 (0.035)*	0.203 (0.046)*

Table 8: Test NRMSE ↓ for different methods. * indicates that the method returned errors during predictions due to learning a stiff ODE.

	Initial conditions $\mathbf{X}_{t_0}$	ODE parameters $\boldsymbol{W}^*$	Learnt Φ
ID-1	$\theta_0 \sim \mathcal{U}(0, \pi/2), \omega_0 = 0$	$\alpha^* \sim \mathcal{U}(1,2), \rho^* \sim \mathcal{U}(0.2, 0.4)$	$\frac{d\theta_t}{dt} = W_1\omega_t$ $\frac{d\omega_t}{dt} = W_2 \sin(\theta_t) + W_3\omega_t$
ID-2	$\theta_0 \sim \mathcal{U}(\pi/4, \pi/2), \omega_0 = 0$	$\alpha^* \sim \mathcal{U}(1,2), \rho^* \sim \mathcal{U}(0.2, 0.4)$	
ID-3	$\theta_0 \sim \mathcal{U}(0, \pi/2), \omega_0 \sim \mathcal{U}(0,1)$	$\alpha^* \sim \mathcal{U}(1,2), \rho^* \sim \mathcal{U}(0.2, 0.4)$	
ID-4	$\theta_0 \sim \mathcal{U}(0, \pi/2), \omega_0 = 0$	$\alpha^* \sim \mathcal{U}(2,3), \rho^* \sim \mathcal{U}(0.2, 0.4)$	
ID-5	$\theta_0 \sim \mathcal{U}(0, \pi/2), \omega_0 = 0$	$\alpha^* \sim \mathcal{U}(1,2), \rho^* \sim \mathcal{U}(0.3, 0.6)$	

Table 9: **(Damped pendulum)** Under various training ranges for initial conditions and parameters, MetaPhysiCa learns the true structure for the damped pendulum system.

We evaluated MetaPhysiCa on a more complex ODE task from Chen (2020) adapted to our setting. We consider a two-dimensional ODE with state $\mathbf{X}_t = [p_t, q_t] \in \mathbb{R}^2$: $\frac{dp_t}{dt} = a^* \sin(p_t) + b^* \sin(q_t^2)$; $\frac{dp_t}{dt} = c^* \sin(p_t)\cos(q_t)$, where $\boldsymbol{W}^* = (a^*, b^*, c^*)$ are the dynamical system parameters. We simulate the ODE over time steps $\{t_0, \ldots, t_T\}$ with $\forall l, t_l = 0.1l, T = 100$ in training and over time steps $\{t_0, \ldots, t_r\}$ in test with $r = \frac{1}{3}T$. In training, we sample initial states $p_t, q_t \sim \mathcal{U}(0.5, 1)$, whereas in out-of-distribution test, we sample $p_t, q_t \sim \mathcal{U}(1, 1.5)$. The dynamical system parameters are sampled as $a^{(i)*}, b^{(i)*}, c^{(i)*} \sim \mathcal{U}(1.0, 1.5)$.

Table 8 shows the results for this task. First, we note that due to the complexity of a 2-layer learnable basis function procedure, we sometimes need to use validation data (held out from training) to cross-validate the learned model (and reject meta-models that do not do well in validation). MetaPhysiCa learnt a stiff ODE for 2 out of 5 folds of cross-validation, resulting in no predictions for in-distribution validation data, which were rejected (marked as superscript $*$). In these experiments MetaPhysiCa performs $1.5\times$ to $1.7\times$ better than the competing baselines. We believe there is room for improvement in the optimization procedure of these more complex models.

D.5 METAPHYSICA WITH DIFFERENT IN- AND OUT-OF-DISTRIBUTION DATA

In this subsection, we evaluate MetaPhysiCa for different in- and out-of-distribution ranges for the initial conditions $\mathbf{X}_{t_0}$ and ODE parameters $\boldsymbol{W}^*$. Table 9 shows that under a variety of training ranges for initial conditions and parameters, MetaPhysiCa is able to learn the true structure of the damped pendulum dynamical system. Furthermore, given the learnt structure, Table 10 shows that MetaPhysiCa outputs robust predictions for different OOD ranges of initial conditions and parameters.

	Initial conditions $\mathbf{X}_{t_0}$	ODE parameters $\boldsymbol{W}^*$	OOD test NRMSE
OOD-1	$\theta_0 \sim \mathcal{U}(\pi - 0.1, \pi), \omega_0 \sim \mathcal{U}(-1, 0)$	$\alpha^* \sim \mathcal{U}(2,3), \rho^* \sim \mathcal{U}(0.4, 0.6)$	0.181 (0.012)
OOD-2	$\theta_0 \sim \mathcal{U}(\pi/2, \pi), \omega_0 = 0$	$\alpha^* \sim \mathcal{U}(2,3), \rho^* \sim \mathcal{U}(0.4, 0.6)$	0.103 (0.016)
OOD-3	$\theta_0 \sim \mathcal{U}(\pi/2, \pi), \omega_0 = 0$	$\alpha^* \sim \mathcal{U}(1,2), \rho^* \sim \mathcal{U}(0.6, 0.8)$	0.029 (0.005)
OOD-4	$\theta_0 \sim \mathcal{U}(\pi - 0.1, \pi), \omega_0 \sim \mathcal{U}(-1, 0)$	$\alpha^* \sim \mathcal{U}(1,2), \rho^* \sim \mathcal{U}(0.6, 0.8)$	0.039 (0.002)

Table 10: **(Damped pendulum)** OOD test performance of MetaPhysiCa for various OOD test ranges for initial conditions and parameters. Using test-time adaptation, MetaPhysiCa is robust to different OOD ranges for the ODE parameters $\boldsymbol{W}^*$.

Methods	Test NRMSE ↓ OOD $\mathbf{X}_{t_0}$	OOD $\mathbf{X}_{t_0}$ and $\boldsymbol{W}^*$
NeuralODE	0.591 (0.119)	0.717 (0.210)
MetaPhysiCa without sine/cosine basis	0.290 (0.137)	0.468 (0.059)
MetaPhysiCa with all basis terms	**0.070 (0.011)**	**0.181 (0.012)**

Table 11: **(Damped pendulum results)** Normalized RMSE ↓ of test predictions from MetaPhysiCa with and without appropriate basis functions. **While MetaPhysiCa without sine/cosine basis terms performs $1.5\times$ to $2\times$ better OOD than the best baseline, it is unable to extrapolate properly and is worse than MetaPhysiCa with these basis terms included.**

D.6 METAPHYSICA WITHOUT APPROPRIATE BASIS TERMS

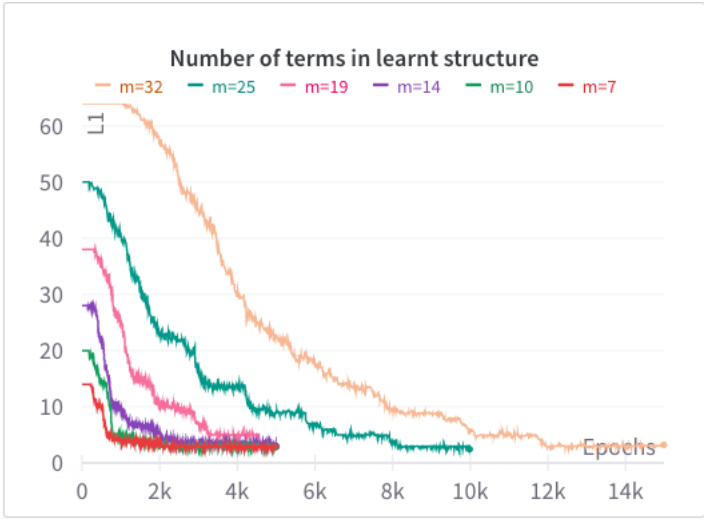
In this subsection, we evaluate MetaPhysiCa without algorithmic alignment, i.e., appropriate basis functions required to learn the ground truth dynamics are not present. We repeat the damped pendulum experiment without sine/cosine basis functions.

In the absence of $\sin(\theta_t)$ term to learn the true dynamics of the damped pendulum system, MetaPhysiCa learns a truncated Taylor series approximation of this term via θ_t and θ_t^3 terms: $\frac{d\theta_t}{dt} = W_1\omega_t$; $\frac{d\omega_t}{dt} = W_2\theta_t + W_3\theta_t^3 + W_4\omega_t$. Table 11 shows that after test-time adaptation, the learnt model achieves $1.5\times$ to $2\times$ better OOD test NRMSE than the best baseline, but is worse than MetaPhysiCa with sinusoidal basis functions included.

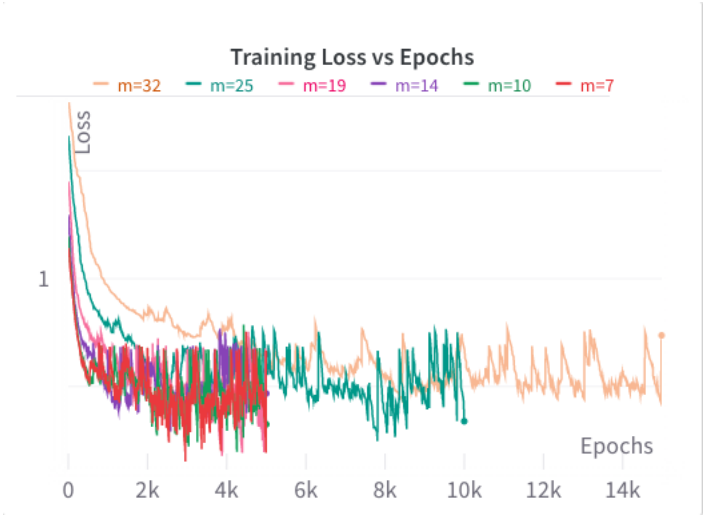
D.7 METAPHYSICA WITH DIFFERENT NUMBER OF BASIS FUNCTIONS

We repeated our damped pendulum experiments with increasing numbers of basis functions. We begin from 7 basis terms (sinusoidal terms and polynomial terms up to power 1) to 32 basis terms (sinusoidal terms and polynomial terms up to power 6) for the two output dimensions in the damped pendulum system (θ_t and ω_t).

Figure 9 shows the loss and $||\hat{\Phi}||_1$ over epoch. Note that when number of basis functions is m, $||\hat{\Phi}||_1$ begins from $2m$ as all basis functions are initialized to be active for both the input dimensions. MetaPhysiCa converges to the true dynamics of the damped pendulum system with 3 basis terms for all the different values of m tested. However, as m increases, model requires a higher number of epochs to reach convergence.



(a) $||\hat{\Phi}||_1$



(b) Training loss

Figure 9: **(Damped pendulum.)** $||\hat{\Phi}||_1$ and training loss vs epochs for different number of basis functions. **MetaPhysiCa learns the true dynamics of the damped pendulum with 3 terms in all the cases; however it takes longer to converge for higher values of m.**