METAPHYSICA: OOD ROBUSTNESS IN PHYSICS-INFORMED MACHINE LEARNING

Anonymous authors
Paper under double-blind review

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. We propose a solution for such tasks, defined as a meta-learning procedure for causal structure discovery. In 3 different OOD tasks, we show that the proposed approach outperforms existing PIML and deep learning methods.

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)) seeks to combine the strengths of physics and machine learning models and has achieved substantial success in tasks where the test data comes from the same distribution as the training data (*in-distribution tasks*). This paper considers an out-of-distribution (OOD) change in the initial system state and unknown parameters of the dynamical system, possibly with different train and test distribution supports (illustrated in Figure 1(a,b)). In this setting, we observe that existing state-of-the-art PIML models perform significantly worse than their performance in-distribution. This is because the standard ML part of PIML, which tends to learn spurious associations, performs poorly in our OOD setting. We then propose a promising solution: Combine *meta learning* with *causal structure discovery* to learn an ODE model that is robust to OOD initial conditions and can adapt to OOD parameters of the dynamical system.

Contributions. Our contributions are: **(i)** 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, **(ii)** *We proposed a hybrid transductive-inductive learning framework for ODEs via meta learning*, where we consider each training and test example/trajectory as separate tasks (transductive), but consider them dependent such that the knowledge can be transferred (inductive), and **(iii)** In order to learn an ODE that is robust to OOD changes in initial conditions (non-overlapping training and test distribution supports), we define a family of structural causal models and perform a structural causal search in order to find the correct model for our task (assumed to be in the family). The proposed method is then empirically validated using three commonly-used simulated physics tasks in OOD scenarios.

2 DYNAMICAL SYSTEM FORECASTING AS A META LEARNING TASK

We formally describe the task of forecasting a dynamical system with a focus on the OOD scenarios.

Definition 1 (Dynamical system forecasting task)**.** *In what follows we describe our task:*

1. **Training data (Figure 1(a)):** *In training, we are given a set of M experiments, which we will denote as M* tasks*. Task $i \in \{1, \dots, M\}$ has an associated (hidden) environment $e^{(i)}$. Different tasks can have the same environment. Let $\mathcal{T}^{(i)} := \mathbf{X}^{(i)}_{t_0}, \dots, \mathbf{X}^{(i)}_{t_{T^{(i)}}}$ denote the noisy observations of our dynamical system, with $\mathbf{X}^{(i)}_t := \boldsymbol{x}^{(i)}_t + \boldsymbol{\varepsilon}^{(i)}_t$, where*

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

$\{t_0, \dots, t_{T^{(i)}}\}$ are discrete time steps, $\boldsymbol{x}^{(i)}_t \in \mathbb{R}^d$ is the (hidden) state of the system at time t during task i, $\boldsymbol{\varepsilon}^{(i)}_t$ are independent zero-mean Gaussian noises, ψ is an unknown deterministic

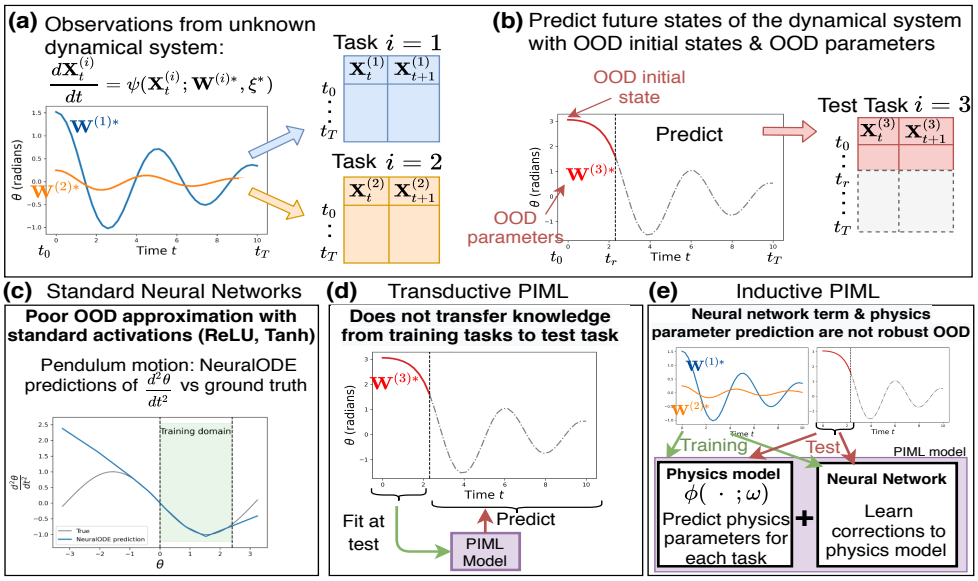


Figure 1: Forecasting problem and traditional approaches to address it. **(a)** Training data consists of multiple observations from the same dynamical system with different parameters $\boldsymbol{W}^{(i)*}$. **(b)** At test, we are given observations till t_r (red solid) and the goal is to predict the future observations till t_T (gray dashed). The initial conditions and the unknown ODE parameters can be OOD in test. **(c)** Shows OOD failure of a standard sequence model for dynamical system forecasting. **(d)** Transductive PIML methods are not able to transfer knowledge from training tasks to a test task with different $\boldsymbol{W}^*$, thus ignore the training data. **(e)** Inductive PIML methods use a neural network to correct a known (incomplete) physics model and another network to inductively predict the physics parameters. These neural networks face OOD robustness issues similar to **(c)**.

function with task-dependent parameters $\boldsymbol{W}^{(i)*}$ and global task-independent parameters $\boldsymbol{\xi}^*$, both hidden. Distribution of initial conditions $\boldsymbol{x}_{t_0}^{(i)} \sim P(\mathbf{X}_{t_0}|E = e^{(i)})$ and of hidden parameters $\boldsymbol{W}^{(i)*} \sim P(\boldsymbol{W}^*|E = e^{(i)})$ for task i may depend on its environment $e^{(i)}$.

2. **Test data (Figure 1(b)):** *At test, we are given noisy observations of the initial sequence $\widetilde{\mathcal{T}}^{(M+1)} := \mathbf{X}_{t_0}^{(M+1)}, \ldots, \mathbf{X}_{t_r}^{(M+1)}$, where r is generally small, of the dynamical system*

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

with initial condition $\boldsymbol{x}_{t_0}^{(M+1)} \sim P(\mathbf{X}_{t_0}|E = e^{(M+1)})$ and (unknown) system parameters $\boldsymbol{W}^{(M+1)*} \sim P(\boldsymbol{W}^*|E = e^{(M+1)})$. **Our task is to predict $\mathbf{X}_{t_{r+1}}^{(M+1)}, \ldots, \mathbf{X}_{t_{T^{(M+1)}}}^{(M+1)}$.**

3. ***OOD initial conditions and system parameters:*** *Initial conditions (resp. ODE parameters) in training can be different from initial conditions (resp. ODE parameters) in test with possibly non-overlapping support due to the presence of a test environment unseen in training.*

In summary, we are given training trajectories that may have (a) different initial conditions, and (b) different unknown ODE system parameters. We observe a test trajectory (indexed by $M + 1$) 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 (unknown) ODE parameters $W^{(M+1)*}$.

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.

Neural network methods. 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 observed in training (Wang et al., 2021a; Geirhos et al., 2020). An MLP's OOD failure can be traced to an absence of appropriate basis and activation functions within the architecture (Xu et al., 2021). Figure 1(c) depicts a similar experiment for dynamical system forecasting using NeuralODE: the model can approximate the 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.*

Physics-informed machine learning (PIML). Several PIML methods have been proposed (e.g., (Daw et al., 2017; Brunton et al., 2016; Yin et al., 2021)) that utilize physics-based domain knowledge for better predictions. To precisely study their OOD challenges, we categorize these methods into inductive and transductive methods based on requirements over ODE parameters $\boldsymbol{W}^*$. In PIML, transductive inference methods treat *training and test examples* as unrelated tasks, hence OOD generalization tends to be less of a challenge in transductive methods. For instance, SINDy (Brunton et al., 2016) and related methods (Martius & Lampert, 2016; Raissi, 2018), learn the ODE equation based on a dictionary of basis functions for a specific parameter $\boldsymbol{W}^{(i)*}$. However, they do not transfer knowledge learnt in training to predict test examples with a different $\boldsymbol{W}^{(j)*}$. This forces these methods to learn only over initial observations of the test task alone, often leading to poor performance (Figure 1(d)). On the other hand, inductive inference focuses on learning rules from the training data that can be applied to unseen test examples, but are fragile OOD since the learned rules are not guaranteed to work outside the training data scope. For example, APHYNITY (Yin et al., 2021) is an inductive method that augments a neural network to a known incomplete physics model where the parameters of the physics model are predicted inductively using a recurrent network. These methods are able to learn from training tasks with different ODE parameters $\boldsymbol{W}^{(i)*}$ (Figure 1(e)). However, the recurrent network fails OOD and often returns incorrect physics parameters OOD and the augmented neural network suffers from the same issues discussed above.

With these key reasons identified for the fragility of existing methods to OOD scenarios, we propose an approach (*MetaPhysiCa*) that outputs more robust predictions OOD.

4 Proposed approach: MetaPhysiCa

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. The causal diagram is depicted in Figure 2 in the plated notation iterating over time.

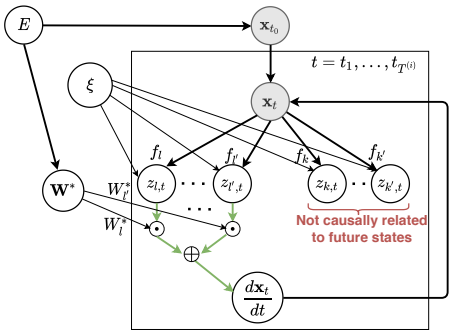


Figure 2: Structural Causal Model

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 $\boldsymbol{x}_t^{(i)} \in \mathbb{R}^d$. Examples of such basis functions include trigonometric functions, polynomial functions, etc. The corresponding outputs from these functions are $z_{k,t}^{(i)} := f_k(\boldsymbol{x}_t^{(i)};\boldsymbol{\xi}_k)$. The derivative ${}^{d\boldsymbol{x}_{t,j}^{(i)}}/_{dt}$ for a particular dimension $j \in \{1, \ldots, d\}$ is only affected by a few (unknown) basis function outputs $z_{k,t}^{(i)}$ (green arrows in Figure 2) and is a linear combination of these selected basis functions with coefficients $\boldsymbol{W}^{(i)*}$. Finally, the derivatives dictate the next state of the dynamical system. We observe the dynamical system with independent additive measurement noise $\mathbf{X}_t^{(i)} := \boldsymbol{x}_t^{(i)} + \boldsymbol{\varepsilon}_t^{(i)}$, where $\boldsymbol{\varepsilon}_t^{(i)} \sim \mathcal{N}(\mathbf{0}, \sigma_\varepsilon^2 I)$. We assume that we are given the collection of m possible basis functions with unknown $\boldsymbol{\xi}$ 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, where we show that appropriate basis functions must be incorporated within the architecture to extrapolate to OOD scenarios.

4.1 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 three-fold: **(a)** discover the true underlying causal structure, i.e., which of the edges $z_{k,t} \to {}^{d\boldsymbol{x}_{t,j}}/_{dt}$ exist for $j = 1, \ldots, d$, **(b)** learn the global parameters $\boldsymbol{\xi}$ that parameterize basis functions, and **(c)** learn the task-specific parameters $\boldsymbol{W}^{(i)*}$ that are coefficients in linear combination of the selected basis functions. 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

$$\frac{d\hat{\mathbf{X}}_t^{(i)}}{dt} = (\boldsymbol{W}^{(i)} \odot \Phi) F(\hat{\mathbf{X}}_t^{(i)};\boldsymbol{\xi}) \ , \tag{2}$$

where $\odot$ is the Hadamard product, $F(\hat{\mathbf{X}}_t^{(i)}; \boldsymbol{\xi}) := \begin{bmatrix} f_1(\hat{\mathbf{X}}_t^{(i)}; \boldsymbol{\xi}_1) & \cdots & f_m(\hat{\mathbf{X}}_t^{(i)}; \boldsymbol{\xi}_m) \end{bmatrix}^T$ are outputs from the basis functions, $\Phi \in \{0,1\}^{d \times m}$ are the learnable parameters governing the global causal structure such that $\Phi_{j,k} = 1$ iff edge $z_{k,t} \to {}^{d\boldsymbol{x}_{t,j}}/_{dt}$ exists in Figure 2, $\boldsymbol{W}^{(i)} \in \mathbb{R}^{d \times m}$ are task-specific parameters that act as coefficients in linear combination of the selected basis functions.

Next we describe a procedure to obtain the structure parameters Φ using a score-based causal structure discovery approach (e.g., Huang et al. (2018)). We wish to find the *minimal* causal structure, i.e., with the least number of edges, that also fits the training data. A sparse structure for Φ implies fewer terms in the RHS of the learnt equation for the derivatives in Equation (2). 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). 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 from integrating Equation (2). However, since the training tasks could have been obtained under different environments (Definition 1), aforementioned approaches based on continuous optimization (e.g., (Zheng et al., 2018)) are not guaranteed to learn the correct structure. They may output a structure that is optimal for an environment with large number of training tasks but suboptimal for others. We use a modified V-REx regularization (Krueger et al., 2021) to learn a structure that minimizes the prediction error across all tasks simultaneously.

Similar to standard meta-learning objectives (Finn et al., 2017; Franceschi et al., 2018), our optimization objective is a bi-level objective that optimizes Φ and ξ in the outer-level, and the task-specific parameters $\boldsymbol{W}^{(i)}$ in the inner-level as follows

$$\hat{\Phi}, \hat{\boldsymbol{\xi}} = \underset{\Phi, \boldsymbol{\xi}}{\arg\min} \frac{1}{M} \sum_{i=1}^{M} R^{(i)}(\hat{\boldsymbol{W}}^{(i)}, \Phi, \boldsymbol{\xi}) + \lambda_\Phi ||\Phi||_1 + \lambda_{\text{REx}} \text{Variance}(\{R^{(i)}(\hat{\boldsymbol{W}}^{(i)}, \Phi, \boldsymbol{\xi})\}_{i=1}^{M})$$
$$\text{s.t.}, \forall i, \hat{\boldsymbol{W}}^{(i)} = \underset{\boldsymbol{W}^{(i)}}{\arg\min} R^{(i)}(\boldsymbol{W}^{(i)}, \Phi, \boldsymbol{\xi}), \tag{3}$$

where λ_Φ and λ_{REx} are hyperparameters. In our experiments, jointly optimizing $\Phi, \boldsymbol{\xi}$ and $\boldsymbol{W}^{(i)}, i = 1, \ldots, M$, resulted in comparable performance with considerable computational benefits instead of alternating SGD (Borkar, 1997). Additional implementation details are provided in Appendix B.

4.2 Transductive test-time adaptation

Finally, given a test task $\widetilde{\mathcal{T}}^{(M+1)} = (\mathbf{X}_{t_0}^{(M+1)}, \ldots, \mathbf{X}_{t_r}^{(M+1)})$ with the unknown ground-truth parameters $\boldsymbol{W}^{(M+1)*} \sim P(\boldsymbol{W}^* | E = e^{(M+1)})$, we adapt the learnt model's task-specific parameters $\boldsymbol{W}^{(M+1)}$ by optimizing the following while keeping $\hat{\Phi}, \hat{\boldsymbol{\xi}}$ fixed

$$\hat{\boldsymbol{W}}^{(M+1)} = \underset{\boldsymbol{W}^{(M+1)}}{\arg\min} \frac{1}{t_r + 1} \sum_{t=t_0}^{t_r} ||\hat{\mathbf{X}}_t^{(M+1)} - \mathbf{X}_t^{(M+1)}||_2^2 \tag{4}$$

where $\hat{\mathbf{X}}_t^{(M+1)} = \mathbf{X}_{t_0}^{(M+1)} + \int_{t_0}^{t} (\boldsymbol{W}^{(M+1)} \odot \hat{\Phi}) F(\hat{\mathbf{X}}_\tau^{(M+1)}; \hat{\boldsymbol{\xi}}) d\tau$ are the predictions obtained using the optimal values $\hat{\Phi}, \hat{\xi}$. Equation (4) allows the model to be robust to OOD ODE parameters.

5 Summary of Results & Conclusions

A detailed description of the experiments is presented in Appendix A, here we summarize the results. We evaluate **MetaPhysiCa** in 3 synthetic forecasting tasks (ODEs) from the literature (Yin et al., 2021; Wang et al., 2021a), namely, Damped pendulum system, Predator-prey system and Epidemic modeling, all adapted to our OOD scenarios with OOD initial conditions $\mathbf{X}_{t_0}$ and OOD ODE parameters $\boldsymbol{W}^*$. Comparing MetaPhysiCa against six baselines spanning standard deep learning (Chen et al., 2018), meta learning (Wang et al., 2021b; Kirchmeyer et al., 2022), and physics-informed machine learning (Brunton et al., 2016; Martius & Lampert, 2016; Yin et al., 2021), **we observe that MetaPhysiCa performs the best OOD across all datasets achieving** $2\times$ **to** $28\times$ **lower OOD errors than the best baseline.** The performance gains stem from two factors: **(i)** The optimal meta-model $\hat{\Phi}$ learns the ground truth ODE for all 3 dynamical systems (Appendix C.1), and **(ii)** task-specific parameters are adapted separately to each OOD test task (Appendix C.2).

Conclusions. We considered the OOD 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 is significantly more robust than the baselines.

REFERENCES

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

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.

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.

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.

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.

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.

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.

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.

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.

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.

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. Multistep neural networks for data-driven discovery of nonlinear dynamical systems. *arXiv preprint arXiv:1801.01236*, 2018.

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.

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.

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.

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: OOD Robustness in Physics-informed Machine Learning"

A EMPIRICAL EVALUATION

We evaluate **MetaPhysiCa** 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 the following approaches: **(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 environment 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 $\sin, \cos$ and other basis functions within a neural network and learns a sparse model. Additional implementation details about the models is presented in Appendix B.

Dataset generation. As per Definition 1, for each dynamical system, we simulate the respective ODE to generate $M = 1000$ training tasks each observed over regularly-spaced discrete time steps $\{t_0, \dots, t_T\}$ [1] where $\forall l, t_l = 0.1l$. For each training task $\mathcal{T}^{(i)}, i = 1, \dots, M$, we sample an initial condition $\mathbf{X}_{t_0}^{(i)} \sim P(\mathbf{X}_{t_0}|E = e)$ where $E = e$ is the training environment. Similarly, we sample different $\boldsymbol{W}^{(i)*} \sim P(\boldsymbol{W}^*|E = e)$ for each training task i. At OOD test, we generate $M' = 200$ test tasks by simulating the respective dynamical system over timesteps $\{t_0, \dots, t_r\}$, where again $\forall l, t_l = 0.1l$. For each test task $j = 1, \dots, M'$, we sample test initial conditions $\mathbf{X}_{t_0}^{(j)} \sim P(\mathbf{X}_{t_0}|E = e')$ and test ODE parameters $\boldsymbol{W}^{(j)*} \sim P(\boldsymbol{W}^*|E = e')$, where $E = e'$ is the test environment. We consider two OOD scenarios: **(a) (OOD $\mathbf{X}_{t_0}$.)** when only the initial conditions are OOD, and **(b) (OOD $\mathbf{X}_{t_0}$ and $\boldsymbol{W}^*$.)** when initial conditions and ODE parameters are OOD. The latter can induce completely different test supports for both the initial conditions and the ODE parameters. The test 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.

Our data generation process is succinctly depicted in Table 1. For each dataset, the second column shows the state variables $\mathbf{X}_t$ and the unknown parameters $\boldsymbol{W}^*$. The three columns "ID", "OOD $\mathbf{X}_{t_0}$" and "OOD $\mathbf{X}_{t_0}$ and $\boldsymbol{W}^*$" depict the distributions of the initial condition and ODE parameters in the respective scenarios. For clarity, we show the values for $\boldsymbol{W}_{\text{param}}$ in the table such that in-distribution $\boldsymbol{W}^{(i)*} \sim \mathcal{U}(\boldsymbol{W}_{\text{param}}, 2\boldsymbol{W}_{\text{param}})$ and out-of-distribution $\boldsymbol{W}^{(i)*} \sim \mathcal{U}(2\boldsymbol{W}_{\text{param}}, 3\boldsymbol{W}_{\text{param}})$.

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 $\{t_0, \dots, t_T\}$ with $\forall l, t_l = 0.1l, T = 100$ in training and over time steps $\{t_0, \dots, t_r\}$ 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 $\{t_0, \dots, t_T\}$ with $\forall l, t_l = 0.1l, T = 100$ in training and over time steps $\{t_0, \dots, t_r\}$ in test with $r = \frac{1}{3}T$. We generate $M = 1000$ training tasks

[1] In our experiments, we let $T^{(i)} = T$ constant for all tasks for simplicity of implementation but the proposed method is not restricted to this case.

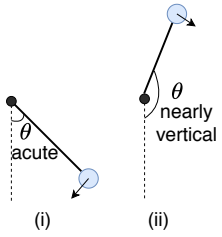
Datasets	State variables	ID	OOD $\mathbf{X}_{t_0}$	OOD $\mathbf{X}_{t_0}$ and $\boldsymbol{W}^*$
Damped pendulum	$\mathbf{X}_t = (\theta_t, \omega_t)$ $\boldsymbol{W}^* = (\alpha, \rho)$	$\theta_0 \sim \mathcal{U}(0, \pi/2)$ $\omega_0 = 0$	$\theta_0 \sim \mathcal{U}(\pi - 0.1, \pi)$ $\omega_0 \sim \mathcal{U}(-1, 0)$ $\alpha_{\text{param}} = 1, \rho_{\text{param}} = 0.2$	$\theta_0 \sim \mathcal{U}(\pi - 0.1, \pi)$ $\omega_0 \sim \mathcal{U}(-1, 0)$
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 1: 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 $\rho^{(i)*} \sim \mathcal{U}(2\rho_{\text{param}}, 3\rho_{\text{param}}) = (0.4, 0.6)$.

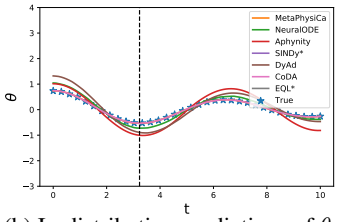
with different initial prey and predator populations with prey $p_{t_0}^{(i)} \sim \mathcal{U}(1000, 2000)$ and predator $q_{t_0}^{(i)} \sim \mathcal{U}(10, 20)$ for each $i = 1, \ldots, M$. At OOD test, we generate $M' = 200$ out-of-distribution (OOD) test tasks with different initial prey populations $p_{t_0}^{(j)} \sim \mathcal{U}(100, 200)$ 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.

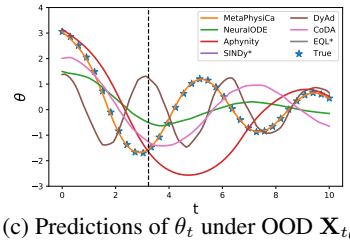
Results. We repeat our experiments 5 times with random seeds and report in-distribution (ID) and out-of-distribution (OOD) normalized root mean squared errors (NRMSE), i.e., RMSE normalized with standard deviation of the ground truth. Figures 3 to 5 show the errors and example predictions from all models for the three datasets respectively. The first column of Tables 3d, 4a, 5a shows in-distribution results while the last two columns show the respective OOD scenarios. 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 3c, 4c and 5c) from these methods show that they have not learnt the true dynamics of the system. For example, for epidemic modeling (Figure 4c), 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 cannot use the training data and are fit on the test observations alone (see Figure 1(d)). Thus, they are unable to identify an accurate analytical equation from these few test observations,



(a) Pendulum task



(b) In-distribution predictions of θ_t



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

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)**

(d) 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.

Figure 3: **(a)** Predict pendulum motion from noisy observations: (i) in-distribution, when dropped from acute angles and (ii) OOD w.r.t initial conditions and parameters, when a different pendulum is dropped from nearly vertical angles. **(b, c)** shows example ground truth curves (blue stars) in- and out-of-distribution along with predictions from different models. While most tested methods perform well in-distribution, only MetaPhysiCa (orange) closely follows the true curve OOD and all other methods are terribly non-robust. **(d)** Standard deep learning methods and physics-informed machine learning methods fail to forecast accurately out-of-distribution. On the other hand, **MetaPhysiCa outputs up to** $4\times$ **more robust OOD predictions.**

resulting in prediction issues due to stiff ODEs. MetaPhysiCa performs the best OOD across all datasets achieving $2\times$ to $28\times$ lower NRMSE OOD errors than the best baseline.

Qualitative analysis. MetaPhysiCa's performance gains stem from two factors: **(i)** The optimal meta-model $\hat{\Phi}$ learns the ground truth ODE (possibly reparameterized) for all 3 dynamical systems (shown in Appendix C.1), 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 (via algorithmic alignment) and the latter helps to be robust over OOD parameters $\boldsymbol{W}^*$. We further show in an ablation study (Appendix C.2) that sparsity regularization (i.e., $||\Phi||_1$) and test-time adaptation (Equation (4)) are the most important components of MetaPhysiCa; OOD performance degrades significantly without either.

B Implementation Details

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

B.1 MetaPhysiCa

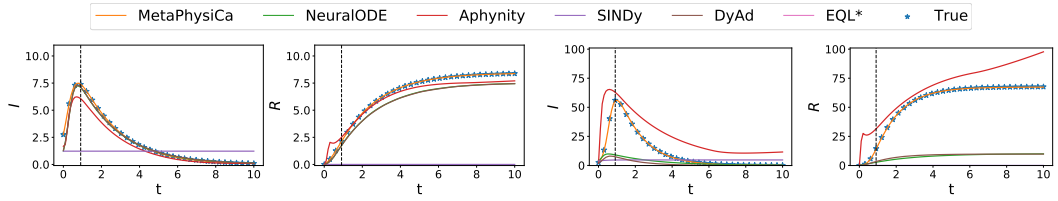
Figure 6 shows a schematic diagram of MetaPhysiCa and the corresponding training/test procedures. Recall from Equation (2) that the proposed model is defined as

$$\frac{d\hat{\mathbf{X}}_t^{(i)}}{dt} = (\boldsymbol{W}^{(i)} \odot \Phi) F(\hat{\mathbf{X}}_t^{(i)}; \boldsymbol{\xi}) \,, \tag{5}$$

where $\odot$ is the Hadamard product and

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) Test NRMSE ↓ for different methods. NaN* indicates that the model returned errors during test.



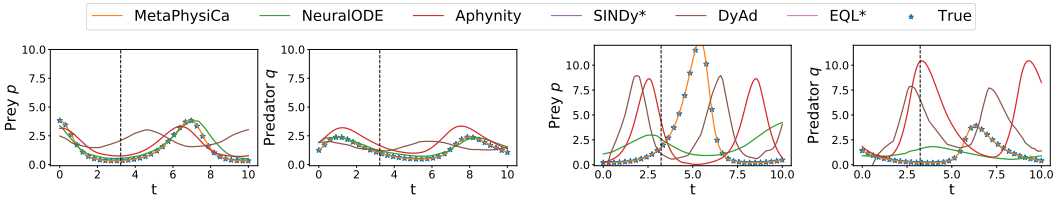
(b) In-distribution predictions

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

Figure 4: **(Epidemic model results) (a) MetaPhysiCa outputs** $28\times$ **and** $9\times$ **more robust OOD predictions** for the two OOD scenarios respectively. **(b, c)** shows example ground truth curves (blue stars) in- and out-of-distribution along with corresponding predictions. Only MetaPhysiCa (orange) closely follows the true curve OOD.

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)**

(a) Test NRMSE ↓ for different methods. NaN* indicates that the model returned errors during test.



(b) In-distribution predictions

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

Figure 5: **(Predator-prey results) (a) MetaPhysiCa outputs** $8\times$ **and** $2\times$ **more robust OOD predictions** in the two OOD scenarios respectively. **(b, c)** shows example ground truth curves (blue stars) in- and out-of-distribution along with corresponding predictions. While most tested methods perform well in-distribution, only MetaPhysiCa (orange) closely follows the true curve OOD.

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

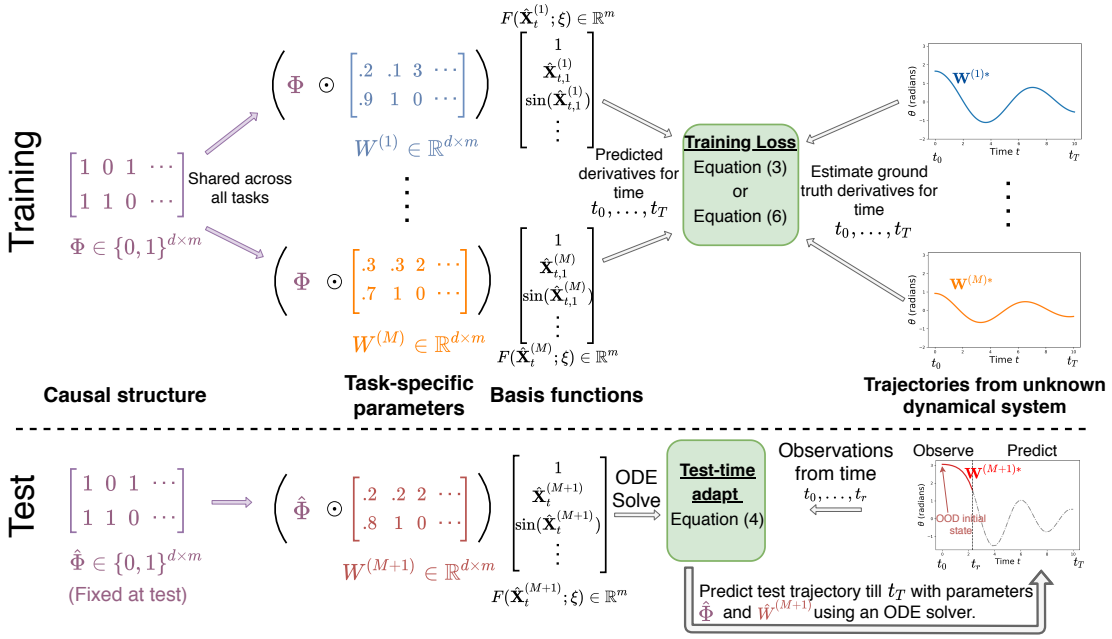


Figure 6: Schematic diagram of MetaPhysiCa and corresponding training/test methodologies. We observe M trajectories in training from the same dynamical system with different initial conditions and ODE parameters. In training, Φ, denoting the causal structure, is shared among all tasks $i = 1, \ldots, M$, while $\boldsymbol{W}^{(i)}$ are the task-specific parameters. Predicted derivatives for task i over time $t = t_0, \ldots, t_T$ are obtained from Equation (2) using the parameters $\Phi, \boldsymbol{W}^{(i)}$ and the basis functions $F(\mathbf{X}_t^{(i)}; \boldsymbol{\xi})$. During test, we adapt $\boldsymbol{W}^{(M+1)}$ over the observations of the test trajectory from time $t_0, \ldots, t_r$, keeping the learnt causal structure $\hat{\Phi}$ fixed.

- $\Phi \in \{0,1\}^{d \times m}$ 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 (3) 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 $\boldsymbol{W}^{(i)}, i = 1, \dots, M$, instead of alternating SGD resulted in comparable performance with considerable computational benefits. We use the following joint optimization objective to approximate Equation (3),

$$\hat{\Phi}, \hat{\boldsymbol{\xi}}, \hat{\boldsymbol{W}}^{(1)}, \dots, \hat{\boldsymbol{W}}^{(M)} = \arg\min_{\Phi, \boldsymbol{\xi}, \boldsymbol{W}^{(1)}, \dots, \boldsymbol{W}^{(M)}} \frac{1}{M} \sum_{i=1}^{M} R^{(i)}(\boldsymbol{W}^{(i)}, \Phi, \boldsymbol{\xi}) + \lambda_\Phi ||\Phi||_1 + \lambda_{\text{REx}} \text{Variance}(\{R^{(i)}(\boldsymbol{W}^{(i)}, \Phi, \boldsymbol{\xi})\}_{i=1}^M) \tag{6}$$

In practice, we use squared loss directly between the predicted and estimated ground truth derivatives instead of $R^{(i)}$, i.e., $\widetilde{R}^{(i)}(\boldsymbol{W}^{(i)}, \Phi, \boldsymbol{\xi}) = \frac{1}{T^{(i)}+1} \sum_{t=t_0}^{t_{T^{(i)}}} ||{}^{d\hat{\mathbf{X}}_t^{(i)}}/_{dt} - {}^{d\mathbf{X}^{(i)}}/_{dt}||_2^2$, which leads to a stable learning procedure with better accuracy in-distribution and OOD. We perform a grid search over the following hyperparameters: regularization strengths $\lambda_\Phi \in \{10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}\}$, $\lambda_{\text{REx}} \in \{0, 10^{-3}, 10^{-2}\}$, and learning rates $\eta \in \{10^{-2}, 10^{-3}, 10^{-4}\}$. 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.

B.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 $\boldsymbol{z}_{\leq r} = F_{\text{enc}}(\mathbf{X}_{t_0}, \dots, \mathbf{X}_{t_r}; \boldsymbol{W}_2)$ 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 $\eta \in \{10^{-2}, 10^{-3}, 10^{-4}\}$.

B.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 $\eta \in \{10^{-2}, 10^{-3}, 10^{-4}\}$.

B.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 $\hat{\Theta}^{(i)}_{\text{phy}} = G_{\text{nn}}(\mathbf{X}_{t_0}, \ldots, \mathbf{X}_{t_r}; \boldsymbol{W}_2)$. Then, APHYNITY augments the given physics model ϕ with a feedforward neural network component F_{nn} and defines the final dynamics as $\frac{d\hat{\mathbf{X}}^{(i)}_t}{dt} = \phi(\hat{\mathbf{X}}^{(i)}_t; \hat{\Theta}^{(i)}_{\text{phy}}) + F_{\text{nn}}(\hat{\mathbf{X}}^{(i)}_t; \boldsymbol{W}_1)$. 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 $\eta \in \{10^{-2}, 10^{-3}, 10^{-4}\}$.

B.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 $\alpha \in \{0.05, 0.01, 0.1, 0.5\}$.

B.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 $\boldsymbol{o}$

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

where f_i are unary basis functions (such as $\sin$, $\cos$, etc.) and g_i are binary basis functions (such as multiplication). We use $\mathrm{id}, \sin$ 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 $\eta \in \{10^{-2}, 10^{-3}, 10^{-4}\}$.

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 2: **(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 (4). **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.**

C ADDITIONAL RESULTS

C.1 QUALITATIVE ANALYSIS

Recall from Equation (2) that the proposed model is defined as

$$\frac{d\hat{\mathbf{X}}_t^{(i)}}{dt} = (\boldsymbol{W}^{(i)} \odot \Phi) F(\hat{\mathbf{X}}_t^{(i)}; \boldsymbol{\xi}) \ , \tag{7}$$

where $F(\hat{\mathbf{X}}_t^{(i)}; \boldsymbol{\xi})$ 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.

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 2 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 (4) 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 5, 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.

C.2 ABLATION RESULTS

We present an ablation study comparing different components of MetaPhysiCa in Table 3. 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.

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 3: **(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, whereas the V-REx penalty (Krueger et al., 2021) helps in some tasks, and performs comparably in others.**