AUGMENTING PHYSICAL MODELS WITH DEEP NETWORKS FOR COMPLEX DYNAMICS FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

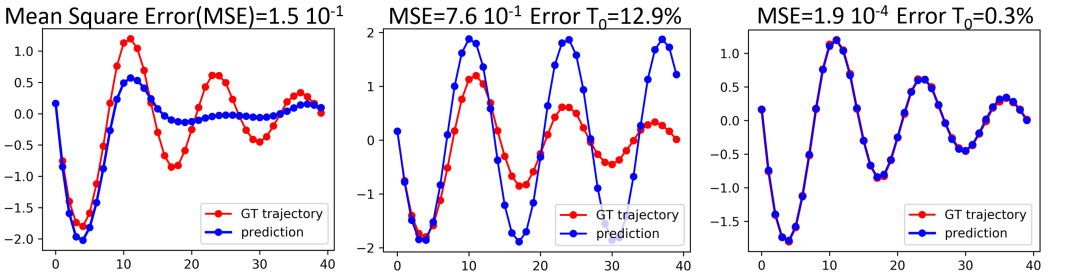
Forecasting complex dynamical phenomena in settings where only partial knowledge of their dynamics is available is a prevalent problem across various scientific fields. While purely data-driven approaches are arguably insufficient in this context, standard physical modeling based approaches tend to be over-simplistic, inducing non-negligible errors. In this work, we introduce the APHYNITY framework, a principled approach for augmenting *incomplete* physical dynamics described by differential equations with deep data-driven models. It consists in decomposing the dynamics into two components: a physical component accounting for the dynamics for which we have some prior knowledge, and a data-driven component accounting for errors of the physical model. The learning problem is carefully formulated such that the physical model explains as much of the data as possible, while the data-driven component only describes information that cannot be captured by the physical model, no more, no less. This not only provides the existence and uniqueness for this decomposition, but also ensures interpretability and benefits generalization. Experiments made on three important use cases, each representative of a different family of phenomena, i.e. reaction-diffusion equations, wave equations and the non-linear damped pendulum, show that APHYNITY can efficiently leverage approximate physical models to accurately forecast the evolution of the system and correctly identify relevant physical parameters.

1 INTRODUCTION

Modeling and forecasting complex dynamical systems is a major challenge in domains such as environment and climate (Rolnick et al., 2019), health science (Choi et al., 2016), and in many industrial applications (Toubeau et al., 2018). Model Based (MB) approaches typically rely on partial or ordinary differential equations (PDE/ODE) and stem from a deep understanding of the underlying physical phenomena. Machine learning (ML) and deep learning methods are more prior agnostic yet have become state-of-the-art for several spatio-temporal prediction tasks (Shi et al., 2015; Wang et al., 2018; Oreshkin et al., 2020), and connections have been drawn between deep architectures and numerical ODE solvers, e.g. neural ODEs (Chen et al., 2018; Ayed et al., 2019b). However, modeling complex physical dynamics is still beyond the scope of pure ML methods, which often cannot properly extrapolate to new conditions as MB approaches do.

Combining the MB and ML paradigms is an emerging trend to develop the interplay between the two paradigms. For example, Brunton et al. (2016); Long et al. (2018b) learn the explicit form of PDEs directly from data, Raissi et al. (2019); Sirignano & Spiliopoulos (2018) use NNs as implicit methods for solving PDEs, Seo et al. (2020) learn spatial differences with a graph network, Ummenhofer et al. (2020) introduce continuous convolutions for fluid simulations, de Bézenac et al. (2018) learn the velocity field of an advection-diffusion system, Greydanus et al. (2019); Chen et al. (2020) enforce conservation laws in the network architecture or in the loss function.

The large majority of aforementioned MB/ML hybrid approaches assume that the physical model adequately describes the observed dynamics. This assumption is, however, commonly violated in practice. This may be due to various factors, e.g. idealized assumptions and difficulty to explain processes from first principles (Gentine et al., 2018), computational constraints prescribing a fine grain modeling of the system (Ayed et al., 2019a), unknown external factors, forces and sources which are present (Large & Yeager, 2004). In this paper, we aim at leveraging prior dynamical



(a) Data-driven Neural ODE (b) Simple physical model (c) Our APHYNITY framework

Figure 1: Predicted dynamics for the damped pendulum vs. ground truth (GT) trajectories ${}^{\mathrm{d}^2\theta}/_{\mathrm{d}t^2} + \omega_0^2 \sin\theta + \alpha {}^{\mathrm{d}\theta}/_{\mathrm{d}t} = 0$. We show that in (a) the data-driven approach (Chen et al., 2018) fails to properly learn the dynamics due to the lack of training data, while in (b) an ideal pendulum cannot take friction into account. The proposed APHYNITY shown in (c) augments the over-simplified physical model in (b) with a data-driven component. APHYNITY improves both forecasting (MSE) and parameter identification (Error T_0) compared to (b).

ODE/PDE knowledge in situations where this physical model is incomplete, i.e. unable to represent the whole complexity of observed data. To handle this case, we introduce a principled learning framework to Augment incomplete PHYsical models for ideNtIfying and forecasTing complex dYnamics (APHYNITY). The rationale of APHYNITY, illustrated in Figure 1 on the pendulum problem, is to *augment* the physical model when—and only when—it falls short.

Designing a general method for combining MB and ML approaches is still a widely open problem, and a clear problem formulation for the latter is lacking (Reichstein et al., 2019). Our contributions towards these goals are the following:

- We introduce a simple yet principled framework for combining both approaches. We decompose the data into a physical and a data-driven term such that the data-driven component only models information that cannot be captured by the physical model. We provide existence and uniqueness guarantees (Section 3.1) for the decomposition given mild conditions, and show that this formulation ensures interpretability and benefits generalization.
- We propose a trajectory-based training formulation (Section 3.2) along with an adaptive optimization scheme (Section 3.3) enabling end-to-end learning for both physical and deep learning components. This allows APHYNITY to *automatically* adjust the complexity of the neural network to different approximation levels of the physical model, paving the way to learning flexible hybrid models.
- We demonstrate the generality of the approach on three use cases (reaction-diffusion, wave equations and the pendulum) representative of different PDE families (parabolic, hyperbolic), having a wide spectrum of application domains, e.g. acoustics, electromagnetism, chemistry, biology, physics (Section 4). We show that APHYNITY can perform similarly to complete physical models by augmenting incomplete ones, both in forecasting accuracy and physical parameter identification. APHYNITY can also be successfully extended to the partially observable setting (see discussion in Section 5).

2 Related work

Correction in data assimilation Prediction under approximate physical models has been tackled by traditional statistical calibration techniques, which often rely on bayesian methods (see the review Pernot & Cailliez, 2017). In data assimilation techniques, e.g. the Kalman filter (Kalman, 1960), 4D-var (Courtier et al., 1994), prediction errors and observation processes are modeled probabilistically and an optimal correction with observed data is applied after each prediction step. Similarly in model-based reinforcement learning, model deficiencies are typically handled by considering only short-term rollouts (Janner et al., 2019) or by model predictive control (Becker et al., 2019). The main originality of APHYNITY is to learn the correction term with deep NNs, and to design a decomposition enabling the physical and correction components to cooperate during training.

Augmented physical models Recently, some attempts have been made to explore the problem of cooperation between physical and data-driven models. HybridNet (Long et al., 2018a) and PhICNet (Saha et al., 2020) both use data-driven networks to learn additive perturbations or source terms to a given PDE. The former considers the favorable context where the perturbations can be accessed, and the latter the special case of additive noise on the input. Wang et al. (2019a); Mehta et al. (2020) propose several empirical fusion strategies with deep nets but they are not grounded on theoretical arguments. PhyDNet (Le Guen & Thome, 2020) tackles augmentation in partially-observed settings, but with specific recurrent architectures dedicated to video prediction. Crucially, all the aforementioned approaches do not address the issues of uniqueness of the decomposition or of proper cooperation for correct parameter identification. Besides, we found experimentally that this vanilla cooperation is inferior to the APHYNITY learning scheme in terms of forecasting and parameter identification performances (see experiments in Section 4.2).

3 THE APHYNITY MODEL

In the following, we study dynamics driven by an equation of the form:

$$\frac{\mathrm{d}X_t}{\mathrm{d}t} = F(X_t) \tag{1}$$

defined over a finite time interval $[0, T]$, where the state X is either vector-valued, i.e. we have $X_t \in \mathbb{R}^d$ for every t, (pendulum equations in Section 4), or X_t is a d-dimensional vector field over a spatial domain $\Omega \subset \mathbb{R}^k$, with $k \in \{2, 3\}$, i.e. $X_t(x) \in \mathbb{R}^d$ for every $(t, x) \in [0, T] \times \Omega$ (reaction-diffusion and wave equations in Section 4). We suppose that we have access to a set of observed trajectories $\mathcal{D} = \{X. : [0, T] \to \mathcal{A} \mid \forall t \in [0, T], {}^{\mathrm{d}X_t}/_{\mathrm{d}t} = F(X_t)\}$, where $\mathcal{A}$ is the set of X values (either $\mathbb{R}^d$ or vector field). In our case, $F : \mathcal{A} \to \mathcal{A}$ is unknown and we only assume that $F \in \mathcal{F}$, with $(\mathcal{F}, \|\cdot\|)$ a normed vector space.

3.1 DECOMPOSING DYNAMICS INTO PHYSICAL AND AUGMENTED TERMS

As introduced in Section 1, we consider the common situation where incomplete information is available on the dynamics, under the form of a family of ODEs or PDEs characterized by their temporal evolution $\mathcal{F}_p \subset \mathcal{F}$. The APHYNITY framework leverages the knowledge of $\mathcal{F}_p$ while mitigating the approximations induced by this simplified model through the combination of physical and data-driven components. $\mathcal{F}$ being a vector space, we can write:

$$F = F_p + F_a$$

where $F_p \in \mathcal{F}_p$ encodes the incomplete physical knowledge and $F_a \in \mathcal{F}$ is the data-driven augmentation term complementing F_p. The incomplete physical prior is supposed to belong to a known family, but the physical parameters (e.g. propagation speed for the wave equation) are unknown and need to be estimated from data. Both F_p and F_a parameters are estimated by fitting the trajectories from $\mathcal{D}$.

The decomposition $F = F_p + F_a$ is in general not unique. For example, all the dynamics could be captured by the F_a component. This decomposition is thus ill-defined, which hampers the interpretability and the extrapolation abilities of the model. In other words, one wants the estimated parameters of F_p to be as close as possible to the true parameter values of the physical model and F_a to play only a complementary role w.r.t F_p, so *as to model only the information that cannot be captured by the physical prior*. For example, when $F \in \mathcal{F}_p$, the data can be fully described by the physical model, and in this case it is sensible to desire F_a to be nullified; this is of central importance in a setting where one wishes to identify physical quantities, and for the model to generalize and extrapolate to new conditions. In a more general setting where the physical model is incomplete, the action of F_a on the dynamics, as measured through its norm, should be as small as possible.

This general idea is embedded in the following optimization problem:

$$\min_{F_p \in \mathcal{F}_p, F_a \in \mathcal{F}} \quad \|F_a\| \quad \text{subject to} \quad \forall X \in \mathcal{D}, \forall t, \frac{\mathrm{d}X_t}{\mathrm{d}t} = (F_p + F_a)(X_t) \tag{2}$$

A first key question is whether the minimum in Eq. (2) is indeed well-defined, in other words whether there exists indeed a decomposition with a minimal norm F_a. The answer actually depends on the geometry of $\mathcal{F}_p$, and is formulated in the following proposition proven in Appendix B:

Proposition 1 (Existence of a minimizing pair)**.** *If $\mathcal{F}_p$ is a proximinal set*[1]*, there exists a decomposition minimizing Eq. (2).*

Proximinality is a mild condition which, as shown through the proof of the proposition, cannot be weakened. It is a property verified by any boundedly compact set. In particular, it is true for closed subsets of finite dimensional spaces. However, if only existence is guaranteed, while forecasts would be expected to be accurate, non-uniqueness of the decomposition would hamper the interpretability of F_p and this would mean that the identified physical parameters are not uniquely determined.

It is then natural to ask under which conditions solving problem Eq. (2) leads to a unique decomposition into a physical and a data-driven component. The following result provides guarantees on the existence and uniqueness of the decomposition under mild conditions. The proof is given in Appendix B:

Proposition 2 (Uniqueness of the minimizing pair)**.** *If $\mathcal{F}_p$ is a Chebyshev set*[1]*, Eq. (2) admits a unique minimizer. The F_p in this minimizer pair is the metric projection of the unknown F onto $\mathcal{F}_p$.*

The Chebyshev assumption condition is strictly stronger than proximinality but si still quite mild and necessary. Indeed, in practice, many sets of interest are Chebyshev, including all closed convex spaces in strict normed spaces and, if $\mathcal{F} = L^2$, $\mathcal{F}_p$ can be any closed convex set, including all finite dimensional subspaces. In particular, all examples considered in the experiments are Chebyshev sets.

Propositions 1 and 2 provide, under mild conditions, the theoretical guarantees for the APHYNITY formulation to infer the correct MB/ML decomposition, thus enabling both recovering the proper physical parameters and accurate forecasting.

3.2 Solving APHYNITY with deep neural networks

In the following, both terms of the decomposition are parametrized and are denoted as $F_p^{\theta_p}$ and $F_p^{\theta_a}$. Solving APHYNITY then consists in estimating the parameters θ_p and θ_a. θ_p are the physical parameters and are typically low-dimensional, e.g. 2 or 3 in our experiments for the considered physical models. For F_a, we need sufficiently expressive models able to optimize over all $\mathcal{F}$: we thus use deep neural networks, which have shown promising performances for the approximation of differential equations (Raissi et al., 2019; Ayed et al., 2019b).

When learning the parameters of $F_p^{\theta_p}$ and $F_a^{\theta_a}$, we have access to a finite dataset of trajectories discretized with a given temporal resolution Δt: $\mathcal{D}_{\text{train}} = \{(X^{(i)}_{k\Delta t})_{0\leq k\leq \lfloor T/\Delta t\rfloor}\}_{1\leq i\leq N}$. Solving Eq. (2) requires estimating the state derivative $\mathrm{d}X_t/\mathrm{d}t$ appearing in the constraint term. One solution is to approximate this derivative using e.g. finite differences as in (Brunton et al., 2016; Greydanus et al., 2019; Cranmer et al., 2020). This numerical scheme requires high space and time resolutions in the observation space in order to get reliable gradient estimates. Furthermore it is often unstable, leading to explosive numerical errors as discussed in Appendix D. We propose instead to solve Eq. (2) using an integral trajectory-based approach: we compute $\widetilde{X}^i_{k\Delta t, X_0}$ from an initial state $X_0^{(i)}$ using the current $F_p^{\theta_p} + F_a^{\theta_a}$ dynamics, then enforce the constraint $\widetilde{X}^i_{k\Delta t, X_0} = X^i_{k\Delta t}$. This leads to our final objective function on (θ_p, θ_a):

$$\min_{\theta_p,\theta_a} \quad \left\|F_a^{\theta_a}\right\| \quad \text{subject to} \quad \forall i, \forall k, \widetilde{X}^{(i)}_{k\Delta t} = X^{(i)}_{k\Delta t} \tag{3}$$

where $\widetilde{X}^{(i)}_{k\Delta t}$ is the approximate solution of the integral $\int_{X_0^{(i)}}^{X_0^{(i)}+k\Delta t} (F_p^{\theta_p} + F_a^{\theta_a})(X_s)\,\mathrm{d}X_s$ obtained by a differentiable ODE solver.

In our setting, where we consider situations for which $F_p^{\theta_p}$ only partially describes the physical phenomenon, this coupled MB + ML formulation leads to different parameter estimates than using the MB formulation alone, as analyzed more thoroughly in Appendix C. Interestingly, our experiments show that using this formulation also leads to a better identification of the physical parameters θ_p than when fitting the simplified physical model $F_p^{\theta_p}$ alone (Section 4). With only an incomplete knowledge

[1] A proximinal set is one from which every point of the space has at least one nearest point. A Chebyshev set is one from which every point of the space has a unique nearest point. More details in Appendix A.

on the physics, θ_p estimator will be biased by the additional dynamics which needs to be fitted in the data. Appendix F also confirms that the integral formulation gives better forecasting results and a more stable behavior than supervising over finite difference approximations of the derivatives.

3.3 ADAPTIVELY CONSTRAINED OPTIMIZATION

The formulation in Eq. (3) involves constraints which are difficult to enforce exactly in practice. We considered a variant of the method of multipliers (Bertsekas, 1996) which uses a sequence of Lagrangian relaxations $\mathcal{L}_{\lambda_j}(\theta_p, \theta_a)$:

$$\mathcal{L}_{\lambda_j}(\theta_p, \theta_a) = \|F_a^{\theta_a}\| + \lambda_j \cdot \mathcal{L}_{traj}(\theta_p, \theta_a) \tag{4}$$

where $\mathcal{L}_{traj}(\theta_p, \theta_a) = \sum_{i=1}^{N} \sum_{h=1}^{T/\Delta t} \|X_{h\Delta t}^{(i)} - \widetilde{X}_{h\Delta t}^{(i)}\|$.

This method needs an increasing sequence $(\lambda_j)_j$ such that the successive minima of $\mathcal{L}_{\lambda_j}$ converge to a solution (at least a local one) of the constrained problem Eq. (3). We select $(\lambda_j)_j$ by using an iterative strategy: starting from a value λ_0, we iterate, minimizing $\mathcal{L}_{\lambda_j}$ by gradient descent[2], then update λ_j with: $\lambda_{j+1} = \lambda_j + \tau_2 \mathcal{L}_{traj}(\theta_{j+1})$, where τ_2 is a chosen hyper-parameter and $\theta = (\theta_p, \theta_a)$. This procedure is summarized in Algorithm 1. This adaptive iterative procedure allows us to obtain stable and robust results, in a reproducible fashion, as shown in the experiments.

Algorithm 1: APHYNITY

Initialization: $\lambda_0 \geq 0, \tau_1 > 0, \tau_2 > 0$;
for *epoch* = $1 : N_{epochs}$ **do**
  for *iter in* $1 : N_{iter}$ **do**
    for *batch in* $1 : B$ **do**
      $\theta_{j+1} = \theta_j - \tau_1 \nabla \left[\lambda_j \mathcal{L}_{traj}(\theta_j) + \|F_a\|\right]$
  $\lambda_{j+1} = \lambda_j + \tau_2 \mathcal{L}_{traj}(\theta_{j+1})$

4 EXPERIMENTAL VALIDATION

We validate our approach on 3 classes of challenging physical dynamics: reaction-diffusion, wave propagation, and the damped pendulum, representative of various application domains such as chemistry, biology or ecology (for reaction-diffusion) and earth physic, acoustic, electromagnetism or even neuro-biology (for waves equations). The two first dynamics are described by PDEs and thus in practice should be learned from very high-dimensional vectors, discretized from the original compact domain. This makes the learning much more difficult than from the one-dimensional pendulum case. For each problem, we investigate the cooperation between physical models of increasing complexity encoding incomplete knowledge of the dynamics (denoted *Incomplete physics* in the following) and data-driven models. We show the relevance of APHYNITY (denoted *APHYNITY models*) both in terms of forecasting accuracy and physical parameter identification.

4.1 EXPERIMENTAL SETTING

We describe the three families of equations studied in the experiments. In all experiments, $\mathcal{F} = \mathcal{L}^2(\mathcal{A})$ where $\mathcal{A}$ is the set of all admissible states for each problem, and the $\mathcal{L}^2$ norm is computed on $\mathcal{D}_{train}$ by: $\|F\|^2 \approx \sum_{i,k} \|F(X_{k\Delta t}^{(i)})\|^2$. All considered sets of physical functionals $\mathcal{F}_p$ are closed and convex in $\mathcal{F}$ and thus are Chebyshev. In order to enable the evaluation on both prediction and parameter identification, all our experiments are conducted on simulated datasets with known model parameters. Each dataset has been simulated using an appropriate high-precision integration scheme for the corresponding equation. All solver-based models take the first state X_0 as input and predict the remaining time-steps by integrating F through the same differentiable generic and common ODE solver (4th order Runge-Kutta)[3]. Implementation details and architectures are given in Appendix E.

Reaction-diffusion equations We consider a FitzHugh-Nagumo type model (Klaasen & Troy, 1984), driven by the PDE $\partial u/\partial t = a\Delta u + R_u(u, v; k), \partial v/\partial t = b\Delta v + R_v(u, v)$ where a, b are the diffusion coefficients, Δ is the Laplace operator. R_u and R_v are local reaction terms. The state is $X = (u, v)$ and is defined over a compact spatial domain Ω with periodic boundary

[2]Convergence to a local minimum isn't necessary, a few steps are often sufficient for a successful optimization.

[3]This integration scheme is then different from the one used for data generation, the rationale for this choice being that when training a model one does not know how exactly the data has been generated.

conditions. The considered physical models are: • *Param PDE* (a, b), without reaction terms: $\mathcal{F}_p = \{F_p^{a,b} : (u, v) \mapsto (a\Delta u, b\Delta v) \mid a, b \in [\epsilon, +\infty) \text{ with } \epsilon > 0\}$ • *Param PDE* (a, b, k): $\mathcal{F}_p = \{F_p^{a,b,k} : (u, v) \mapsto (a\Delta u + R_u(u, v; k), b\Delta v + R_v(u, v)) \mid a, b, k \in [\epsilon, +\infty) \text{ with } \epsilon > 0\}$.

Damped wave equations We investigate the damped-wave PDE: ${}^{\partial^2 w}/_{\partial t^2} - c^2\Delta w + k{}^{\partial w}/_{\partial t} = 0$ where k is the damping coefficient. The state is $X = (w, {}^{\partial w}/_{\partial t})$ and we consider a compact spatial domain Ω with Neumann homogeneous boundary conditions. Note that this damping differs from the pendulum, as its effect is global. Our physical models are: • *Param PDE (c)*, without damping term: $\mathcal{F}_p = \{F_p^c : (u, v) \mapsto (v, c^2\Delta u) \mid c \in [\epsilon, +\infty) \text{ with } \epsilon > 0\}$; • *Param PDE* (c, k): $\mathcal{F}_p = \{F_p^{c,k} : (u, v) \mapsto (v, c^2\Delta u - kv) \mid c, k \in [\epsilon, +\infty) \text{ with } \epsilon > 0\}$.

Damped pendulum The evolution follows the ODE ${}^{\mathrm{d}^2\theta}/_{\mathrm{d}t^2} + \omega_0^2 \sin\theta + \alpha{}^{\mathrm{d}\theta}/_{\mathrm{d}t} = 0$, where $\theta(t)$ is the angle, ω_0 the proper pulsation (T_0 the period) and α the damping coefficient. With state $X = (\theta, {}^{\mathrm{d}\theta}/_{\mathrm{d}t})$, the ODE is $F_p^{\omega_0,\alpha} : X \mapsto ({}^{\mathrm{d}\theta}/_{\mathrm{d}t}, -\omega_0^2 \sin\theta - \alpha{}^{\mathrm{d}\theta}/_{\mathrm{d}t})$. Our physical models are: • *Hamiltonian* (Greydanus et al., 2019), a conservative approximation, with $\mathcal{F}_p = \{F_p^{\mathcal{H}} : (u, v) \mapsto (\partial_y \mathcal{H}(u, v), -\partial_x \mathcal{H}(u, v)) \mid \mathcal{H} \in H^1(\mathbb{R}^2)\}$, $H^1(\mathbb{R}^2)$ is the first order Sobolev space. • *Param ODE* (ω_0), the frictionless pendulum: $\mathcal{F}_p = \{F_p^{\omega_0,\alpha=0} \mid \omega_0 \in [\epsilon, +\infty) \text{ with } \epsilon > 0\}$ • *Param ODE* (ω_0, α), the full pendulum equation: $\mathcal{F}_p = \{F_p^{\omega_0,\alpha} \mid \omega_0, \alpha \in [\epsilon, +\infty) \text{ with } \epsilon > 0\}$.

Baselines As purely data-driven baselines, we use Neural ODE (Chen et al., 2018) for the three problems and PredRNN++ (Wang et al., 2018, for reaction-diffusion only) which are competitive models for datasets generated by differential equations and for spatio-temporal data. As MB/ML methods, in the ablations studies (see Appendix F), we compare for all problems, to the vanilla MB/ML cooperation scheme found in (Wang et al., 2019a; Mehta et al., 2020). We also show results for *True PDE/ODE*, which corresponds to the equation for data simulation (which do not lead to zero error due to the difference between simulation and training integration schemes). For the pendulum, we compare to Hamiltonian neural networks (Greydanus et al., 2019; Toth et al., 2020) and to the the deep Galerkin method (DGM, Sirignano & Spiliopoulos, 2018). See additional details in Appendix E.

4.2 Results

We analyze and discuss below the results obtained for the three kind of dynamics. We successively examine different evaluation or quality criteria. The conclusions are consistent for the three problems, which allows us to highlight clear trends for all of them.

Forecasting accuracy The data-driven models do not perform well compared to *True PDE/ODE* (all values in $\log$ MSE): -4.6 for PredRNN++ vs. -9.17 for reaction-diffusion, -2.51 vs. -5.24 for wave equation, and -2.84 vs. -8.44 for the pendulum in Table 1. The Deep Galerkin method for the pendulum in complete physics *DGM* (ω_0, α), being constrained by the equation, outperforms Neural ODE but is far inferior to APHYNITY models. In the incomplete physics case, *DGM* (ω_0) fails to compensate for the missing information. The *incomplete physical models*, *Param PDE* (a, b) for the reaction-diffusion, *Param PDE (c)* for the wave equation, and *Param ODE* (ω_0) and *Hamiltonian models* for the damped pendulum, have even poorer performances than purely data-driven ones, as can be expected since they ignore important dynamical components, e.g. friction in the pendulum case. Using APHYNITY with these imperfect physical models greatly improves forecasting accuracy in all cases, significantly outperforming purely data-driven models, and reaching results often close to the accuracy of the true ODE, when APHYNITY and the true ODE models are integrated with the same numerical scheme (which is different from the one used for data generation, hence the non-null errors even for the true equations), e.g. -5.92 vs. -5.24 for wave equation in Table 1. This clearly highlights the capacity of our approach to augment incomplete physical models with a learned data-driven component.

Physical parameter estimation Confirming the phenomenon mentioned in the introduction and detailed in Appendix C, incomplete physical models can lead to bad estimates for the relevant physical parameters: an error respectively up to 67.6% and 10.4% for parameters in the reaction-diffusion and wave equations, and an error of more than 13% for parameters for the pendulum in Table 1. APHYNITY is able to significantly improve physical parameters identification: 2.3% error for the reaction-diffusion, 0.3% for the wave equation, and 4% for the pendulum. This validates the fact that augmenting a simple physical model to compensate its approximations is not only beneficial for

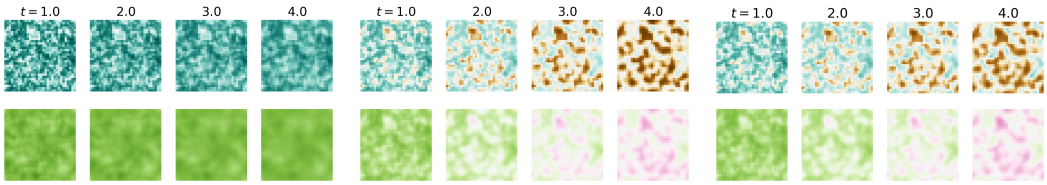
Table 1: Forecasting and identification results on the (a) reaction-diffusion, (b) wave equation, and (c) damped pendulum datasets. We set for (a) $a = 1 \times 10^{-3}, b = 5 \times 10^{-3}, k = 5 \times 10^{-3}$, for (b) $c = 330$, $k = 50$ and for (c) $T_0 = 6$, $\alpha = 0.2$ as true parameters. log MSEs are computed respectively over 25, 25, and 40 predicted time-steps. %Err param. averages the results when several physical parameters are present. For each level of incorporated physical knowledge, equivalent best results according to a Student t-test are shown in bold. n/a corresponds to non-applicable cases.

Dataset		Method	log MSE	%Err param.	$\|F_a\|^2$
(a) Reaction-diffusion	Data-driven	Neural ODE	-3.76±0.02	n/a	n/a
		PredRNN++	-4.60±0.01	n/a	n/a
	Incomplete physics	Param PDE (a, b)	-1.26±0.02	67.6	n/a
		APHYNITY Param PDE (a, b)	**-5.10±0.21**	**2.3**	67
	Complete physics	Param PDE (a, b, k)	**-9.34±0.20**	0.17	n/a
		APHYNITY Param PDE (a, b, k)	**-9.35±0.02**	**0.096**	1.5e-6
		True PDE	-8.81±0.05	n/a	n/a
		APHYNITY True PDE	**-9.17±0.02**	n/a	1.4e-7
(b) Wave equation	Data-driven	Neural ODE	-2.51±0.29	n/a	n/a
	Incomplete physics	Param PDE (c)	0.51±0.07	10.4	n/a
		APHYNITY Param PDE (c)	**-4.64±0.25**	**0.3**	71.
	Complete physics	Param PDE (c, k)	-4.68±0.55	1.38	n/a
		APHYNITY Param PDE (c, k)	**-6.09±0.28**	**0.63**	4.35
		True PDE	-4.66±0.30	n/a	n/a
		APHYNITY True PDE	**-5.24±0.45**	n/a	0.14
(c) Damped pendulum	Data-driven	Neural ODE	-2.84±0.70	n/a	n/a
	Incomplete physics	Hamiltonian	-0.35±0.10	n/a	n/a
		APHYNITY Hamiltonian	**-3.97±1.20**	n/a	623
		Param ODE (ω_0)	-0.14±0.10	13.2	n/a
		Deep Galerkin Method (ω_0)	-3.10±0.40	22.1	n/a
		APHYNITY Param ODE (ω_0)	**-7.86±0.60**	**4.0**	132
	Complete physics	Param ODE (ω_0, α)	**-8.28±0.40**	**0.45**	n/a
		Deep Galerkin Method (ω_0, α)	-3.14±0.40	7.1	n/a
		APHYNITY Param ODE (ω_0, α)	**-8.31±0.30**	**0.39**	8.5
		True ODE	**-8.58±0.20**	n/a	n/a
		APHYNITY True ODE	**-8.44±0.20**	n/a	2.3

prediction, but also helps to limit errors for parameter identification when dynamical models do not fit data well. This is crucial for interpretability and explainability of the estimates.

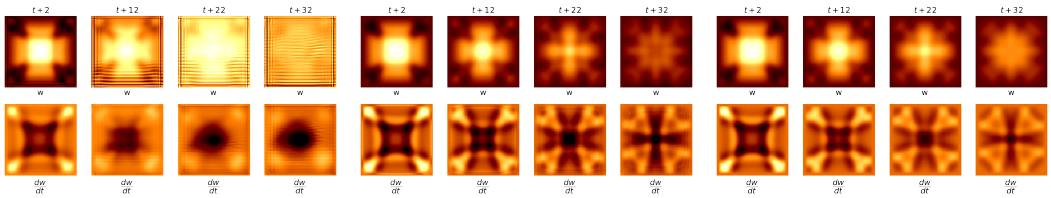
Ablation study We conduct ablation studies to validate the importance of the APHYNITY augmentation compared to a naive strategy consisting in learning $F = F_p + F_a$ without taking care on the quality of the decomposition, as done in (Wang et al., 2019a; Mehta et al., 2020). Results shown in Table 1 of Appendix F show a consistent gain of APHYNITY for the three use cases and for all physical models: for instance for *Param ODE* (a, b) in reaction-diffusion, both forecasting performances (log MSE =-5.10 vs. -4.56) and identification parameter (Error= 2.33% vs. 6.39%) improve. Other ablation results are provided in Appendix F showing the relevance of the the trajectory-based approach described in Section 3.2 (vs supervising over finite difference approximations of the derivative F).

Flexibility When applied to complete physical models, APHYNITY does not degrade accuracy, contrary to a vanilla cooperation scheme (see ablations in Appendix F). This is due to the least action principle of our approach: when the physical knowledge is sufficient for properly predicting the observed dynamics, the model learns to ignore the data-driven augmentation. This is shown by the norm of the trained neural net component F_a, which is reported in Table 1 last column: as expected, $\|F_a\|^2$ diminishes as the complexity of the corresponding physical model increases, and becomes almost null for complete physical models. We see that the norm of F_a is a good indication of how imperfect the physical models $\mathcal{F}_p$ are. It highlights the flexibility of APHYNITY to successfully adapt to very different levels of prior knowledge. Note also that APHYNITY sometimes slightly improves over the true ODE, because it compensates the error introduced by different numerical integration methods for data simulation and training (see Appendix E).



(a) Param PDE (a, b), diffusion-only (b) APHYNITY Param PDE (a, b) (c) Ground truth simulation

Figure 2: Comparison of predictions of two components u (top) and v (bottom) of the reaction-diffusion system. Note that $t = 4$ is largely beyond the dataset horizon ($t = 2.5$).



(a) Neural ODE (b) APHYNITY Param PDE (c) (c) Ground truth simulation

Figure 3: Comparison between the prediction of APHYNITY when c is estimated and Neural ODE for the damped wave equation. Note that $t + 32$, last column for (a, b, c) is already beyond the training time horizon ($t + 25$), showing the consistency of APHYNITY method.

Qualitative visualizations Results in Figure 2 for reaction-diffusion show that the incomplete diffusion parametric PDE in Figure 2(a) is unable to properly match ground truth simulations: the behavior of the two components in Figure 2(a) is reduced to simple independent diffusions due to the lack of interaction terms between u and v. By using APHYNITY in Figure 2(b), the correlation between the two components appears together with the formation of Turing patterns, which is very similar to the ground truth. This confirms that F_a can learn the reaction terms and improve prediction quality. In Figure 3, we see for the wave equation that the data-driven Neural ODE model fails at approximating $\mathrm{d}w/\mathrm{d}t$ as the forecast horizon increases: it misses crucial details for the second component $\mathrm{d}w/\mathrm{d}t$ which makes the forecast diverge from the ground truth. APHYNITY incorporates a Laplacian term as well as the data-driven F_a thus capturing the damping phenomenon and succeeding in maintaining physically sound results for long term forecasts, unlike Neural ODE.

Extension to non-stationary dynamics We provide additional results in Appendix G to tackle datasets where physical parameters of the equations vary in each sequence. To this end, we design an encoder able to perform parameter estimation for each sequence. Results show that APHYNITY accommodates well to this setting, with similar trends as those reported in this section.

5 Discussion and Conclusion

In this work, we introduce the APHYNITY framework that can efficiently augment approximate physical models with deep data-driven networks, performing similarly to models for which the underlying dynamics are entirely known. We exhibit the superiority of APHYNITY over data-driven, incomplete physics, and state-of-the-art approaches combining ML and MB methods, both in terms of forecasting and parameter identification on three various classes of physical systems. Besides, APHYNITY is flexible enough to adapt to different approximation levels of prior physical knowledge.

An appealing perspective is the applicability of APHYNITY on partially-observable settings. As a first step towards this goal, we evaluate APHYNITY on the Moving MNIST video prediction task (Srivastava et al., 2015). We show that we can reach new state-of-the-art performances on this dataset (see Appendix H Table 10), by using the APHYNITY augmentation framework on top on the recent PhyDNet architecture (Le Guen & Thome, 2020). It shows that the MB/ML cooperation brought up by APHYNITY can be successfully leveraged in a latent space for improving performances. We hope that the APHYNITY framework will open up the way to the design of a wide range of more flexible MB/ML models, e.g. in climate science, robotics or reinforcement learning. In particular, analyzing the theoretical decomposition properties in a partially-observed setting is an important direction for future work.

REFERENCES

Ibrahim Ayed, Nicolas Cedilnik, Patrick Gallinari, and Maxime Sermesant. Ep-net: Learning cardiac electrophysiology models for physiology-based constraints in data-driven predictions. In Yves Coudière, Valéry Ozenne, Edward J. Vigmond, and Nejib Zemzemi (eds.), *Functional Imaging and Modeling of the Heart - 10th International Conference, FIMH 2019, Bordeaux, France, June 6-8, 2019, Proceedings*, volume 11504 of *Lecture Notes in Computer Science*, pp. 55–63. Springer, 2019a.

Ibrahim Ayed, Emmanuel de Bézenac, Arthur Pajot, Julien Brajard, and Patrick Gallinari. Learning dynamical systems from partial observations. *arXiv preprint arXiv:1902.11136*, 2019b.

Philipp Becker, Harit Pandya, Gregor Gebhardt, Cheng Zhao, James Taylor, and Gerhard Neumann. Recurrent kalman networks: Factorized inference in high-dimensional deep feature spaces. *International Conference on Machine Learning (ICML)*, 2019.

Dimitri P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods (Optimization and Neural Computation Series)*. Athena Scientific, 1 edition, 1996.

Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems (NeurIPS)*, pp. 6571–6583, 2018.

Zhengdao Chen, Jianyu Zhang, Martin Arjovsky, and Léon Bottou. Symplectic recurrent neural networks. *International Conference on Learning Representations (ICLR)*, 2020.

Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3504–3512, 2016.

Philippe Courtier, J-N Thépaut, and Anthony Hollingsworth. A strategy for operational implementation of 4d-var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, 120(519):1367–1387, 1994.

Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. *ICLR 2020 Deep Differential Equations Workshop*, 2020.

Emmanuel de Bézenac, Arthur Pajot, and Patrick Gallinari. Deep learning for physical processes: Incorporating prior scientific knowledge. *International Conference on Learning Representations (ICLR)*, 2018.

John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

James Fletcher and Warren Moors. Chebyshev sets. *Journal of the Australian Mathematical Society*, 98:161–231, 04 2014. doi: 10.1017/S1446788714000561.

P. Gentine, M. Pritchard, S. Rasp, G. Reinaudi, and G. Yacalis. Could machine learning break the convection parameterization deadlock? *Geophysical Research Letters*, 45(11):5742–5751, 2018.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 15353–15363, 2019.

Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li F Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 517–526, 2018.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 12519–12530, 2019.

Gordon G Johnson. A nonconvex set which has the unique nearest point property. *Journal of Approximation Theory*, 51(4):289 – 332, 1987.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

Gene A. Klaasen and William C. Troy. Stationary wave solutions of a system of reaction-diffusion equations derived from the fitzhugh–nagumo equations. *SIAM Journal on Applied Mathematics*, 44(1):96–110, 1984. doi: 10.1137/0144008.

William Large and Stephen Yeager. Diurnal to decadal global forcing for ocean and sea-ice models: The data sets and flux climatologies, 05 2004.

Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Computer Vision and Pattern Recognition (CVPR)*. 2020.

Yun Long, Xueyuan She, and Saibal Mukhopadhyay. Hybridnet: integrating model-based and data-driven learning to predict evolution of dynamical systems. *Conference on Robot Learning (CoRL)*, 2018a.

Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. PDE-Net: Learning PDEs from data. In *International Conference on Machine Learning (ICML)*, 2018b.

Viraj Mehta, Ian Char, Willie Neiswanger, Youngseog Chung, and Jeff Schneider. Neural dynamical systems. *ICLR 2020 Deep Differential Equations Workshop*, 2020.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *International Conference on Learning Representations (ICLR)*, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. 2019.

Pascal Pernot and Fabien Cailliez. A critical review of statistical calibration/prediction models handling data inconsistency and model inadequacy. *AIChE Journal*, 63(10):4642–4665, 2017.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 473:686–707, 2019.

Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, and & Prabhat. Deep learning and process understanding for data-driven Earth system science. *Nature*, 566:195–204, 2019.

David Rolnick, Priya L Donti, Lynn H Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, et al. Tackling climate change with machine learning. In *NeurIPS 2019 workshop on Climate Change with Machine Learning*, 2019.

Priyabrata Saha, Saurabh Dash, and Saibal Mukhopadhyay. PHICNet: Physics-incorporated convolutional recurrent neural networks for modeling dynamical systems. *arXiv preprint arXiv:2004.06243*, 2020.

Sungyong Seo, Chuizheng Meng, and Yan Liu. Physics-aware difference graph networks for sparsely-observed dynamics. *International Conference on Learning Representations (ICLR)*, 2020.

Xingjian Shi, 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 (NeurIPS)*, pp. 802–810, 2015.

Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using LSTMs. In *International Conference on Machine Learning (ICML)*, pp. 843–852, 2015.

Peter Toth, Danilo Jimenez Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. *International Conference on Learning Representations (ICLR)*, 2020.

Jean-François Toubeau, Jérémie Bottieau, François Vallée, and Zacharie De Grève. Deep learning-based multivariate probabilistic forecasting for short-term scheduling in power markets. *IEEE Transactions on Power Systems*, 34(2):1203–1215, 2018.

Benjamin Ummenhofer, Lukas Prantl, Nils Thuerey, and Vladlen Koltun. Lagrangian fluid simulation with continuous convolutions. *International Conference on Learning Representations (ICLR)*, 2020.

Qi Wang, Feng Li, Yi Tang, and Yan Xu. Integrating model-driven and data-driven methods for power system frequency stability assessment and control. *IEEE Transactions on Power Systems*, 34(6):4557–4568, 2019a.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 879–888. 2017.

Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and Philip S. Yu. PredRNN++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *International Conference on Machine Learning (ICML)*, 2018.

Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3D LSTM: A model for video prediction and beyond. In *International Conference on Learning Representations (ICLR)*, 2019b.

Yunbo Wang, Jianjin Zhang, Hongyu Zhu, Mingsheng Long, Jianmin Wang, and Philip S Yu. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 9154–9162, 2019c.

A Reminder on proximinal and Chebyshev sets

We begin by giving a definition of proximinal and Chebyshev sets, taken from (Fletcher & Moors, 2014):

Definition 1. *A proximinal set of a normed space $(E, \|\cdot\|)$ is a subset $\mathcal{C} \subset E$ such that every $x \in E$ admits at least a nearest point in $\mathcal{C}$.*

Definition 2. *A Chebyshev set of a normed space $(E, \|\cdot\|)$ is a subset $\mathcal{C} \subset E$ such that every $x \in E$ admits a unique nearest point in $\mathcal{C}$.*

Proximinality reduces to a compacity condition in finite dimensional spaces. In general, it is a weaker one: Boundedly compact sets verify this property for example.

In Euclidean spaces, Chebyshev sets are simply the closed convex subsets. The question of knowing whether it is the case that all Chebyshev sets are closed convex sets in infinite dimensional Hilbert spaces is still an open question. In general, there exists examples of non-convex Chebyshev sets, a famous one being presented in (Johnson, 1987) for a non-complete inner-product space.

Given the importance of this topic in approximation theory, finding necessary conditions for a set to be Chebyshev and studying the properties of those sets have been the subject of many efforts. Some of those properties are summarized below:

- The metric projection on a boundedly compact Chebyshev set is continuous.
- If the norm is strict, every closed convex space, in particular any finite dimensional subspace is Chebyshev.
- In a Hilbert space, every closed convex set is Chebyshev.

B Proof of Propositions 1 and 2

We prove the following result which implies both propositions in the article:

Proposition 3. *The optimization problem:*

$$\min_{F_p \in \mathcal{F}_p, F_a \in \mathcal{F}} \quad \|F_a\| \quad \text{subject to} \quad \forall X \in \mathcal{D}, \forall t, \frac{\mathrm{d}X_t}{\mathrm{d}t} = (F_p + F_a)(X_t) \tag{5}$$

is equivalent a metric projection onto $\mathcal{F}_p$.

If $\mathcal{F}_p$ is proximinal, Eq. (5) admits a minimizing pair.

If $\mathcal{F}_p$ is Chebyshev, Eq. (5) admits a unique minimizing pair which F_p is the metric projection.

Proof. The idea is to reconstruct the full functional from the trajectories of $\mathcal{D}$. By definition, $\mathcal{A}$ is the set of points reached by trajectories in $\mathcal{D}$ so that:

$$\mathcal{A} = \{x \in \mathbb{R}^d \mid \exists X_\cdot \in \mathcal{D}, \exists t,\ X_t = x\}$$

Then let us define a function $F^{\mathcal{D}}$ in the following way: For $a \in \mathcal{A}$, we can find $X_\cdot \in \mathcal{D}$ and t_0 such that $X_{t_0} = a$. Differentiating X at t_0, which is possible by definition of $\mathcal{D}$, we take:

$$F^{\mathcal{D}}(a) = \left.\frac{\mathrm{d}X_t}{\mathrm{d}t}\right|_{t=t_0}$$

For any (F_p, F_a) satisfying the constraint in Eq. (5), we then have that $(F_p + F_a)(a) = {}^{\mathrm{d}X_t}/_{\mathrm{d}t|t_0} = F^{\mathcal{D}}(a)$ for all $a \in \mathcal{A}$. Conversely, any pair such that $(F_p, F_a) \in \mathcal{F}_p \times \mathcal{F}$ and $F_p + F_a = F^{\mathcal{D}}$, verifies the constraint.

Thus we have the equivalence between Eq. (5) and the metric projection formulated as:

$$\underset{F_p \in \mathcal{F}_p}{\text{minimize}} \quad \left\|F^{\mathcal{D}} - F_p\right\| \tag{6}$$

If $\mathcal{F}_p$ is proximinal, the projection problem admits a solution which we denote $F_p^\star$. Taking $F_a^\star = F^{\mathcal{D}} - F_p^\star$, we have that $F_p^\star + F_a^\star = F^{\mathcal{D}}$ so that $(F_p^\star, F_a^\star)$ verifies the constraint of Eq. (2). Moreover, if there is (F_p, F_a) satisfying the constraint of Eq. (2), we have that $F_p + F_a = F^{\mathcal{D}}$ by what was shown above and $\|F_a\| = \|F^{\mathcal{D}} - F_p\| \geq \|F^{\mathcal{D}} - F_p^\star\|$ by definition of $F_p^\star$. This shows that $(F_p^\star, F_a^\star)$ is minimal.

Moreover, if $\mathcal{F}_p$ is a Chebyshev set, by uniqueness of the projection, if $F_p \neq F_p^\star$ then $\|F_a\| > \|F_a^\star\|$. Thus the minimal pair is unique.

□

C PARAMETER ESTIMATION IN INCOMPLETE PHYSICAL MODELS

Classically, when a set $\mathcal{F}_p \subset \mathcal{F}$ summarising the most important properties of a system is available, this gives a *simplified model* of the true dynamics and the adopted problem is then to fit the trajectories using this model as well as possible, solving:

$$\begin{aligned} \underset{F_p \in \mathcal{F}_p}{\text{minimize}} \quad & \mathbb{E}_{X\sim\mathcal{D}} L(\widetilde{X}^{X_0}, X) \\ \text{subject to} \quad & \forall g \in \mathcal{I},\ \widetilde{X}_0^g = g \text{ and } \forall t,\ \frac{\mathrm{d}\widetilde{X}_t^g}{\mathrm{d}t} = F_p(\widetilde{X}_t^g) \end{aligned} \tag{7}$$

where L is a discrepancy measure between trajectories. Recall that $\widetilde{X}^{X_0}$ is the result trajectory of an ODE solver taking X_0 as initial condition. In other words, we try to find a function F_p which gives trajectories as close as possible to the ones from the dataset. While estimation of the function becomes easier, there is then a residual part which is left unexplained and this can be a non negligible issue in at least two ways:

- When $F \notin \mathcal{F}_p$, the loss is strictly positive at the minimum. This means that reducing the space of functions $\mathcal{F}_p$ makes us lose in terms of accuracy.[4]
- The obtained function F_p might not even be the most meaningful function from $\mathcal{F}_p$ as it would try to capture phenomena which are not explainable with functions in $\mathcal{F}_p$, thus giving the wrong bias to the calculated function. For example, if one is considering a dampened periodic trajectory where only the period can be learned in $\mathcal{F}_p$ but not the dampening, the estimated period will account for the dampening and will thus be biased.

This is confirmed in the paper in Section 4: the incomplete physical models augmented with APHYNITY get different and experimentally better physical identification results than the physical models alone.

Let us compare our approach with this one on the linearized damped pendulum to show how estimates of physical parameters can differ. The equation is the following:

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} + \omega_0^2\theta + \alpha\frac{\mathrm{d}\theta}{\mathrm{d}t} = 0$$

We take the same notations as in the article and parametrize the simplified physical models as:

$$F_p^a : X \mapsto (\frac{\mathrm{d}\theta}{\mathrm{d}t}, -a\theta)$$

where $a > 0$ corresponds to ω_0^2. The corresponding solution for an initial state X_0, which we denote X^a, can then written explicitly as:

$$\theta_t^a = \theta_0 \cos\sqrt{a}t$$

Let us consider damped pendulum solutions X written as:

$$\theta_t = \theta_0 e^{-t}\cos t$$

which corresponds to:

$$F : X \mapsto (\frac{\mathrm{d}\theta}{\mathrm{d}t}, -2(\theta + \frac{\mathrm{d}\theta}{\mathrm{d}t}))$$

[4] This is true in theory, although not necessarily in practice when F overfits a small dataset.

It is then easy to see that the estimate of a with the physical model alone can be obtained by minimizing:

$$\int_0^T |e^{-t}\cos t - \cos\sqrt{a}t|^2$$

This expression depends on T and thus, depending on the chosen time interval and the way the integral is discretized will almost always give biased estimates. In other words, the estimated value of a will not give us the desired solution $t \mapsto \cos t$.

On the other hand, for a given a, in the APHYNITY framework, the residual must be equal to:

$$F_r^a : X \mapsto (0, (a-2)\theta - 2\frac{\mathrm{d}\theta}{\mathrm{d}t})$$

in order to satisfy the fitting constraint. Here a corresponds to $1+\omega_0^2$ not to ω_0^2 as in the simplified case. Minimizing its norm, we obtain $a = 2$ which gives us the desired solution:

$$\theta_t = \theta_0 e^{-t}\cos t$$

with the right period.

D Discussion on supervision over derivatives

In order to find the appropriate decomposition (F_p, F_a), we use a trajectory-based error by solving:

$$\begin{aligned} \underset{F_p \in \mathcal{F}_p, F_a \in \mathcal{F}}{\text{minimize}} \quad & \|F_a\| \\ \text{subject to} \quad & \forall g \in \mathcal{I},\ \widetilde{X}_0^g = g \text{ and } \forall t,\ \frac{\mathrm{d}\widetilde{X}_t^g}{\mathrm{d}t} = (F_p + F_a)(\widetilde{X}_t^g), \\ & \forall X \in \mathcal{D},\ L(X, \widetilde{X}^{X_0}) = 0 \end{aligned} \tag{8}$$

In the continuous setting where the data is available at all times t, this problem is in fact equivalent to the following one:

$$\underset{F_p \in \mathcal{F}_p}{\text{minimize}} \quad \mathbb{E}_{X\sim\mathcal{D}} \int \left\| \frac{\mathrm{d}X_t}{\mathrm{d}t} - F_p(X_t) \right\| \tag{9}$$

where the supervision is done directly over derivatives, obtained through finite-difference schemes. This echoes the proof in Section B of the Appendix where F can be reconstructed from the continuous data.

However, in practice, data is only available at discrete times with a certain time resolution. While Eq. (9) is indeed equivalent to Eq. (8) in the continuous setting, in the practical discrete one, the way error propagates is not anymore: For Eq. (8) it is controlled over integrated trajectories while for Eq. (9) the supervision is over the approximate derivatives of the trajectories from the dataset. We argue that the trajectory-based approach is more flexible and more robust for the following reasons:

- In Eq. (8), if F_a is appropriately parameterized, it is possible to perfectly fit the data trajectories at the sampled points.
- The use of finite differences schemes to estimate F as is done in Eq. (9) necessarily induces a non-zero discretization error.
- This discretization error is explosive in terms of divergence from the true trajectories.

This last point is quite important, especially when time sampling is sparse (even though we do observe this adverse effect empirically in our experiments with relatively finely time-sampled trajectories). The following gives a heuristical reasoning as to why this is the case. Let $\widetilde{F} = F + \epsilon$ be the function estimated from the sampled points with an error ϵ such that $\|\epsilon\|_\infty \leq \alpha$. Denoting $\widetilde{X}$ the corresponding trajectory generated by $\widetilde{F}$, we then have, for all $X \in \mathcal{D}$:

$$\forall t,\ \frac{\mathrm{d}(X - \widetilde{X})_t}{\mathrm{d}t} = F(X_t) - F(\widetilde{X}_t) - \epsilon(\widetilde{X}_t)$$

Integrating over $[0, T]$ and using the triangular inequality as well as the mean value inequality, supposing that F has uniformly bounded spatial derivatives:

$$\forall t \in [0, T],\ \|(X - \widetilde{X})_t\| \leq \|\nabla F\|_\infty \int_0^t \|X_s - \widetilde{X}_s\| + \alpha t$$

which, using a variant of the Grönwall lemma, gives us the inequality:

$$\forall t \in [0, T],\ \|X_t - \widetilde{X}_t\| \leq \frac{\alpha}{\|\nabla F\|_\infty}(\exp(\|\nabla F\|_\infty t) - 1)$$

When α tends to 0, we recover the true trajectories X. However, as α is bounded away from 0 by the available temporal resolution, this inequality gives a rough estimate of the way $\widetilde{X}$ diverges from them, and it can be an equality in many cases. This exponential behaviour explains our choice of a trajectory-based optimization.

E Implementation details

We describe here the three use cases studied in the paper for validating APHYNITY. All experiments are implemented with PyTorch (Paszke et al., 2019) and the differentiable ODE solvers with the adjoint method implemented in `torchdiffeq`.[5]

E.1 Reaction-diffusion equations

The system is driven by a FitzHugh-Nagumo type PDE (Klaasen & Troy, 1984)

$$\frac{\partial u}{\partial t} = a\Delta u + R_u(u, v; k), \frac{\partial v}{\partial t} = b\Delta v + R_v(u, v)$$

where a and b are respectively the diffusion coefficients of u and v, Δ is the Laplace operator. The local reaction terms are $R_u(u, v; k) = u - u^3 - k - v, R_v(u, v) = u - v$.

The state $X = (u, v)$ is defined over a compact rectangular domain $\Omega = [-1, 1]^2$ with periodic boundary conditions. Ω is spatially discretized with a 32×32 2D uniform square mesh grid. The periodic boundary condition is implemented with circular padding around the borders. Δ is systematically estimated with a 3×3 discrete Laplace operator.

Dataset Starting from a randomly sampled initial state $X_{\text{init}} \in [0, 1]^{2\times32\times32}$, we generate states by integrating the true PDE with fixed a, b, and k in a dataset ($a = 1\times10^{-3}, b = 5\times10^{-3}, k = 5\times10^{-3}$). We firstly simulate high time-resolution ($\delta t_{\text{sim}} = 0.001$) sequences with explicit finite difference method. We then extract states every $\delta t_{\text{data}} = 0.1$ to construct our low time-resolution datasets.

We set the time of random initial state to $t = -0.5$ and the time horizon to $t = 2.5$. 1920 sequences are generated, with 1600 for training/validation and 320 for test. We take the state at $t = 0$ as X_0 and predict the sequence until the horizon (equivalent to 25 time steps) in all reaction-diffusion experiments. Note that the sub-sequence with $t < 0$ are reserved for the extensive experiments in Appendix G.1.

Neural network architectures Our F_a here is a 3-layer convolution network (ConvNet). The two input channels are (u, v) and two output ones are $(\frac{\partial u}{\partial t}, \frac{\partial v}{\partial t})$. The purely data-driven Neural ODE uses such ConvNet as its F. The detailed architecture is provided in Table 2. The estimated physical parameters θ_p in F_p are simply a trainable vector $(a, b) \in \mathbb{R}_+^2$ or $(a, b, k) \in \mathbb{R}_+^3$.

[5] https://github.com/rtqichen/torchdiffeq

Table 2: ConvNet architecture in reaction-diffusion and wave equation experiments, used as data-driven derivative operator in APHYNITY and Neural ODE (Chen et al., 2018).

Module	Specification
2D Conv.	3×3 kernel, 2 input channels, 16 output channels, 1 pixel zero padding
2D Batch Norm.	No average tracking
ReLU activation	—
2D Conv.	3×3 kernel, 16 input channels, 16 output channels, 1 pixel zero padding
2D Batch Norm.	No average tracking
ReLU activation	—
2D Conv.	3×3 kernel, 16 input channels, 2 output channels, 1 pixel zero padding

Optimization hyperparameters We choose to apply the same hyperparameters for all the reaction-diffusion experiments: $Niter = 1, \lambda_0 = 1, \tau_1 = 1 \times 10^{-3}, \tau_2 = 1 \times 10^3$.

E.2 Wave equations

The damped wave equation is defined by

$$\frac{\partial^2 w}{\partial t^2} - c^2 \Delta w + k \frac{\partial w}{\partial t} = 0$$

where c is the wave speed and k is the damping coefficient. The state is $X = (w, \frac{\partial w}{\partial t})$.

We consider a compact spatial domain Ω represented as a 64×64 grid and discretize the Laplacian operator similarly. Δ is implemented using a 5×5 discrete Laplace operator in simulation whereas in the experiment is a 3×3 Laplace operator. Null Neumann boundary condition are imposed for generation.

Dataset δt was set to 0.001 to respect Courant number and provide stable integration. The simulation was integrated using a 4th order finite difference Runge-Kutta scheme for 300 steps from an initial Gaussian state, i.e for all sequence at $t = 0$, we have:

$$w(x, y, t = 0) = C \times \exp^{\frac{(x-x_0)^2 + (y-y_0)^2}{\sigma^2}} \tag{10}$$

The amplitude C is fixed to 1, and $(x_0, y_0) = (32, 32)$ to make the Gaussian curve centered for all sequences. However, σ is different for each sequence and uniformly sampled in $[10, 100]$. The same δt was used for train and test. All initial conditions are Gaussian with varying amplitudes. 250 sequences are generated, 200 are used for training while 50 are reserved as a test set. In the main paper setting, $c = 330$ and $k = 50$. As with the reaction diffusion case, the algorithm takes as input a state $X_{t_0} = (w, \frac{\mathrm{d}w}{\mathrm{d}t})(t_0)$ and predicts all states from $t_0 + \delta t$ up to $t_0 + 25\delta t$.

Neural network architectures The neural network for F_a is a 3-layer convolution neural network with the same architecture as in Table 2. For F_p, the parameter(s) to be estimated is either a scalar $c \in \mathbb{R}_+$ or a vector $(c, k) \in \mathbb{R}_+^2$. Similarly, Neural ODE networks are build as presented in Table 2.

Optimization hyperparameters We use the same hyperparameters for the experiments: $Niter = 3, \lambda_0 = 1, \tau_1 = 1 \times 10^{-4}, \tau_2 = 1 \times 10^2$.

E.3 Damped pendulum

We consider the non-linear damped pendulum problem, governed by the ODE

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} + \omega_0^2 \sin\theta + \alpha \frac{\mathrm{d}\theta}{\mathrm{d}t} = 0$$

where $\theta(t)$ is the angle, $\omega_0 = \frac{2\pi}{T_0}$ is the proper pulsation (T_0 being the period) and α is the damping coefficient. With the state $X = (\theta, \frac{\mathrm{d}\theta}{\mathrm{d}t})$, the ODE can be written as $\frac{\mathrm{d}X_t}{\mathrm{d}t} = F(X_t)$ with $F : X \mapsto (\frac{\mathrm{d}\theta}{\mathrm{d}t}, -\omega_0^2 \sin\theta - \alpha \frac{\mathrm{d}\theta}{\mathrm{d}t})$.

Dataset For each train / validation / test split, we simulate a dataset with 25 trajectories of 40 timesteps (time interval $[0, 20]$, timestep $\delta t = 0.5$) with fixed ODE coefficients $(T_0 = 12, \alpha = 0.2)$ and varying initial conditions. The simulation integrator is Dormand-Prince Runge-Kutta method of order (4)5 (DOPRI5, Dormand & Prince, 1980). We also add a small amount of white gaussian noise ($\sigma = 0.01$) to the state. Note that our pendulum dataset is much more challenging than the ideal frictionless pendulum considered in Greydanus et al. (2019).

Neural network architectures We detail in Table 3 the neural architectures used for the damped pendulum experiments. All data-driven augmentations for approximating the mapping $X_t \mapsto F(X_t)$ are implemented by multi-layer perceptrons (MLP) with 3 layers of 200 neurons and ReLU activation functions (except at the last layer: linear activation). The Hamiltonian (Greydanus et al., 2019; Toth et al., 2020) is implemented by a MLP that takes the state X_t and outputs a scalar estimation of the Hamiltonian $\mathcal{H}$ of the system: the derivative is then computed by an in-graph gradient of $\mathcal{H}$ with respect to the input: $F(X_t) = \left(\frac{\partial \mathcal{H}}{\partial(\mathrm{d}\theta/\,\mathrm{d}t)}, -\frac{\partial \mathcal{H}}{\mathrm{d}\theta}\right)$.

Table 3: Neural network architectures for the damped pendulum experiments. n/a corresponds to non-applicable cases.

Method	Physical model	Data-driven model
Neural ODE	n/a	MLP(in=2, units=200, layers=3, out=2)
Hamiltonian	MLP(in=2, units=200, layers=3, out=1)	n/a
APHYNITY Hamiltonian	MLP(in=2, units=200, layers=3, out=1)	MLP(in=2, units=200, layers=3, out=2)
Param ODE (ω_0)	1 trainable parameter ω_0	n/a
APHYNITY Param ODE (ω_0)	1 trainable parameter ω_0	MLP(in=2, units=200, layers=3, out=2)
Param ODE (ω_0, α)	2 trainable parameters ω_0, λ	n/a
APHYNITY Param ODE (ω_0, α)	2 trainable parameters ω_0, λ	MLP(in=2, units=200, layers=3, out=2)

Optimization hyperparameters The hyperparameters of the APHYNITY optimization algorithm $(Niter, \lambda_0, \tau_1, \tau_2)$ were cross-validated on the validation set and are shown in Table 4. All models were trained with a maximum number of 5000 steps with early stopping.

Table 4: Hyperparameters of the damped pendulum experiments.

Method	Niter	λ_0	τ_1	τ_2
APHYNITY Hamiltonian	5	1	1	0.1
APHYNITY ParamODE (ω_0)	5	1	1	10
APHYNITY ParamODE (ω_0, λ)	5	1000	1	100

F Ablation Study

We conduct ablation studies to show the effectiveness of APHYNITY's adaptive optimization and trajectory-based learning scheme.

F.1 Ablation to vanilla MB/ML cooperation

In Table 5, we consider the ablation case with the vanilla augmentation scheme found in Le Guen & Thome (2020); Wang et al. (2019a); Mehta et al. (2020), which does not present any proper decomposition guarantee. We observe that the APHYNITY cooperation scheme outperforms this vanilla scheme in all case, both in terms of forecasting performances (e.g. log MSE= -0.35 vs. -3.97 for the Hamiltonian in the pendulum case) and parameter identification (e.g. Err Param=8.4% vs. 2.3 for Param PDE (a, b for reaction-diffusion). It confirms the crucial benefits of APHYNITY's principled decomposition scheme.

Table 5: Ablation study comparing APHYNITY to the vanilla augmentation scheme (Wang et al., 2019a; Mehta et al., 2020) for the reaction-diffusion equation, wave equation and damped pendulum.

Dataset	Method	log MSE	%Err Param.	$\|F_a\|^2$
Reaction-diffusion	Param. PDE (a, b) with vanilla aug.	-4.56±0.52	8.4	(7.5±1.4)e1
	APHYNITY Param. PDE (a, b)	**-5.10±0.21**	**2.3**	(6.7±0.4)e1
	Param. PDE (a, b, k) with vanilla aug.	-8.04±0.03	25.4	(1.5±0.2)e-2
	APHYNITY Param. PDE (a, b, k)	**-9.35±0.02**	**0.096**	(1.5±0.4)e-6
	True PDE with vanilla aug.	-8.12±0.05	n/a	(6.1±2.3)e-4
	APHYNITY True PDE	**-9.17±0.02**	n/a	(1.4±0.8)e-7
Wave equation	Param PDE (c) with vanilla aug.	-3.90 ± 0.27	0.51	88.66
	APHYNITY Param PDE (c)	**-4.64±0.21**	**0.31**	71.0
	Param PDE (c, k) with vanilla aug.	-5.96 ± 0.10	0.71	25.1
	APHYNITY Param PDE (c, k)	**-6.09±0.28**	**0.70**	4.54
Damped pendulum	Hamiltonian with vanilla aug.	-0.35±0.1	n/a	837±117
	APHYNITY Hamiltonian	**-3.97±1.2**	n/a	623±68
	Param ODE (ω_0) with vanilla aug.	-7.02±1.7	4.5	148±49
	APHYNITY Param ODE (ω_0)	**-7.86±0.6**	**4.0**	132±11
	Param ODE (ω_0, α) with vanilla aug.	-7.60±0.6	4.65	35.5±6.2
	APHYNITY Param ODE (ω_0, α)	**-8.31±0.3**	**0.38**	8.5±2.0
	Augmented True ODE with vanilla aug.	**-8.40±0.2**	n/a	3.4±0.8
	APHYNITY True ODE	**-8.44±0.2**	n/a	2.3±0.4

F.2 Detailed ablation study

We conduct also two other ablations in Table 6:

- *derivative supervision*: in which $F_p + F_a$ is trained with supervision over approximated derivatives on ground truth trajectory, as performed in Greydanus et al. (2019); Cranmer et al. (2020). More precisely, APHYNITY's $\mathcal{L}_{\text{traj}}$ is here replaced with $\mathcal{L}_{\text{deriv}} = \|\frac{\mathrm{d}X_t}{\mathrm{d}t} - F(X_t)\|$ as in Eq. (9), where $\frac{\mathrm{d}X_t}{\mathrm{d}t}$ is approximated by finite differences on X_t.
- *non-adaptive optim.*: in which we train APHYNITY by minimizing $\|F_a\|$ without the adaptive optimization of λ shown in Algorithm 1. This case is equivalent to $\lambda = 1, \tau_2 = 0$.

We highlight the importance to use a principled adaptive optimization algorithm (APHYNITY algorithm described in paper) compared to a non-adpative optimization: for example in the reaction-diffusion case, log MSE= -4.55 vs. -5.10 for Param PDE (a, b). Finally, when the supervision occurs on the derivative, both forecasting and parameter identification results are systematically lower than with APHYNITY's trajectory based approach: for example, log MSE=-1.16 vs. -4.64 for Param PDE (c) in the wave equation. It confirms the good properties of the APHYNITY training scheme.

Table 6: Detailed ablation study on supervision and optimization for the reaction-diffusion equation, wave equation and damped pendulum.

Dataset	Method	log MSE	%Err Param.	$\|F_a\|^2$
Reaction-diffusion	Augmented Param. PDE (a, b) derivative supervision	-4.42±0.25	12.6	(6.8±0.6)e1
	Augmented Param. PDE (a, b) non-adaptive optim.	-4.55±0.11	7.5	(7.6±1.0)e1
	APHYNITY Param. PDE (a, b)	**-5.10±0.21**	**2.3**	(6.7±0.4)e1
	Augmented Param. PDE (a, b, k) derivative supervision	-4.90±0.06	11.7	(1.9±0.3)e-1
	Augmented Param. PDE (a, b, k) non-adaptive optim.	-9.10±0.02	0.21	(5.5±2.9)e-7
	APHYNITY Param. PDE (a, b, k)	**-9.35±0.02**	**0.096**	(1.5±0.4)e-6
	Augmented True PDE derivative supervision	-6.03±0.01	n/a	(3.1±0.8)e-3
	Augmented True PDE non-adaptive optim.	-9.01±0.01	n/a	(1.5±0.8)e-6
	APHYNITY True PDE	**-9.17±0.02**	n/a	(1.4±0.8)e-7
Wave equation	Augmented Param PDE (c) derivative supervision	-1.16±0.48	12.1	0.00024
	Augmented Param PDE (c) non-adaptive optim.	-2.57±0.21	3.1	43.6
	APHYNITY Param PDE (c)	**-4.64±0.21**	**0.31**	71.0
	Augmented Param PDE (c, k) derivative supervision	-4.19±0.36	7.2	0.00012
	Augmented Param PDE (c, k) non-adaptive optim.	-4.93±0.51	1.32	0.054
	APHYNITY Param PDE (c, k)	**-6.09±0.28**	**0.70**	4.54
	Augmented True PDE non-adaptive optim.	-4.97±0.49	n/a	0.23
	APHYNITY True PDE	**-5.24±0.45**	n/a	0.14
Damped pendulum	Augmented Hamiltonian derivative supervision	-0.83±0.3	n/a	642±121
	Augmented Hamiltonian non-adaptive optim.	-0.49±0.58	n/a	165±30
	APHYNITY Hamiltonian	**-3.97±1.2**	n/a	623±68
	Augmented Param ODE (ω_0) derivative supervision	-1.02±0.04	5.8	136±13
	Augmented Param ODE (ω_0) non-adaptive optim.	-4.30±1.3	4.4	90.4±27
	APHYNITY Param ODE (ω_0)	**-7.86±0.6**	**4.0**	132±11
	Augmented Param ODE (ω_0, α) derivative supervision	-2.61±0.2	5.0	3.2±1.7
	Augmented Param ODE (ω_0, α) non-adaptive optim.	-7.69±1.3	1.65	4.8±7.7
	APHYNITY Param ODE (ω_0, α)	**-8.31±0.3**	**0.38**	8.5±2.0
	Augmented True ODE derivative supervision	-2.14±0.3	n/a	4.1±0.6
	Augmented True ODE non-adaptive optim.	**-8.34±0.4**	n/a	1.4±0.3
	APHYNITY True ODE	**-8.44±0.2**	n/a	2.3±0.4

G ADDITIONAL EXPERIMENTS

G.1 REACTION-DIFFUSION SYSTEMS WITH VARYING DIFFUSION PARAMETERS

We conduct an extensive evaluation on a setting with varying diffusion parameters for reaction-diffusion equations. The only varying parameters are diffusion coefficients, i.e. individual a and b for each sequence. We randomly sample $a \in [1 \times 10^{-3}, 2 \times 10^{-3}]$ and $b \in [3 \times 10^{-3}, 7 \times 10^{-3}]$. k is still fixed to 5×10^{-3} across the dataset.

In order to estimate a and b for each sequence, we use here a ConvNet encoder E to estimate parameters from 5 reserved frames ($t < 0$). The architecture of the encoder E is similar to the one in Table 2 except that E takes 5 frames (10 channels) as input and E outputs a vector of estimated $(\tilde{a}, \tilde{b})$ after applying a sigmoid activation scaled by 1×10^{-2} (to avoid possible divergence). For the baseline Neural ODE, we concatenate a and b to each sequence as two channels.

In Table 7, we observe that combining data-driven and physical components outperforms the pure data-driven one. When applying APHYNITY to Param PDE (a, b), the prediction precision is significantly improved (log MSE: -1.32 vs. -4.32) with a and b respectively reduced from 55.6% and 54.1% to 11.8% and 18.7%. For complete physics cases, the parameter estimations are also improved for Param PDE (a, b, k) by reducing over 60% of the error of b (3.10 vs. 1.23) and 10% to 20% of the errors of a and k (resp. 1.55/0.59 vs. 1.29/0.39).

The extensive results reflect the same conclusion as shown in the main article: APHYNITY improves the prediction precision and parameter estimation. The same decreasing tendency of $\|F_a\|$ is also confirmed.

Table 7: Results of the dataset of reaction-diffusion with varying (a, b). $k = 5 \times 10^{-3}$ is shared across the dataset.

	Method	log MSE	%Err a	%Err b	%Err k	$\|F_a\|^2$
Data-driven	Neural ODE (Chen et al., 2018)	-3.61±0.07	n/a	n/a	n/a	n/a
Incomplete physics	Param PDE (a, b)	-1.32±0.02	55.6	54.1	n/a	n/a
	APHYNITY Param PDE (a, b)	**-4.32±0.32**	**11.8**	**18.7**	n/a	(4.3±0.6)e1
Complete physics	Param PDE (a, b, k)	**-5.54±0.38**	1.55	3.10	0.59	n/a
	APHYNITY Param PDE (a, b, k)	**-5.72±0.25**	**1.29**	**1.23**	**0.39**	(5.9±4.3)e-1
	True PDE	**-8.86±0.02**	n/a	n/a	n/a	n/a
	APHYNITY True PDE	**-8.82±0.15**	n/a	n/a	n/a	(1.8±0.6)e-5

G.2 Additional results for the wave equation

We conduct an experiment where each sequence is generated with a different wave celerity. This dataset is challenging because both c and the initial conditions vary across the sequences. For each simulated sequence, an initial condition is sampled as described previously, along with a wave celerity c also sampled uniformly in $[300, 400]$. Finally our initial state is integrated with the same Runge-Kutta scheme. 200 of such sequences are generated for training while 50 are kept for testing.

For this experiment, we also use a ConvNet encoder to estimate the wave speed c from 5 consecutive reserved states $(w, \frac{\partial w}{\partial t})$. The architecture of the encoder E is the same as in Table 2 but with 10 input channels. Here also, k is fixed for all sequences and $k = 50$. The hyper-parameters used in these experiments are the same than described in the Section E.2.

The results when multiple wave speeds c are in the dataset are consistent with the one present when only one is considered. Indeed, while prediction performances are slightly hindered, the parameter estimation remains consistent for both c and k. This extension provides elements attesting for the robustness and adaptability of our method to more complex settings. Finally the purely data-driven Neural-ODE fails to cope with the increasing difficulty.

Table 8: Results for the damped wave equation when considering multiple c sampled uniformly in $[300, 400]$ in the dataset, k is shared across all sequences and $k = 50$.

	Method	log MSE	%Error c	%Error k	$\|F_a\|^2$
Data-driven	Neural ODE	0.056±0.34	n/a	n/a	n/a
Incomplete physics	Param PDE (c)	-1.32±0.27	23.9	n/a	n/a
	APHYNITY Param PDE (c)	**-4.51±0.38**	3.2	n/a	171
Complete physics	Param PDE (c, k)	-4.25±0.28	3.54	1.43	n/a
	APHYNITY Param PDE (c, k)	**-4.84±0.57**	2.41	0.064	3.64
	True PDE (c, k)	**-4.51±0.29**	n/a	n/a	n/a
	APHYNITY True PDE (c, k)	**-4.49±0.22**	n/a	n/a	0.0005

G.3 Damped pendulum with varying parameters

To extend the experiments conducted in the paper (section 4) with fixed parameters ($T_0 = 6, \alpha = 0.2$) and varying initial conditions, we evaluate APHYNITY on a much more challenging dataset where we vary both the parameters (T_0, α) and the initial conditions between trajectories.

We simulate 500/50/50 trajectories for the train/valid/test sets integrated with DOPRI5. For each trajectory, the period T_0 (resp. the damping coefficient α) are sampled uniformly in the range $[3, 10]$ (resp. $[0, 0.5]$).

We train models that take the first 20 steps as input and predict the next 20 steps. To account for the varying ODE parameters between sequences, we use an encoder that estimates the parameters based

on the first 20 timesteps. In practice, we use a recurrent encoder composed of 1 layer of 128 GRU units. The output of the encoder is fed as additional input to the data-driven augmentation models and to an MLP with final softplus activations to estimate the physical parameters when necessary ($\omega_0 \in \mathbb{R}_+$ for Param ODE (ω_0), $(\omega_0, \alpha) \in \mathbb{R}_+^2$ for Param ODE (ω_0, α)).

In this varying ODE context, we also compare to the state-of-the-art univariate time series forecasting method N-Beats (Oreshkin et al., 2020).

Results shown in Table 9 are consistent with those presented in the paper. Pure data-driven models Neural ODE (Chen et al., 2018) and N-Beats (Oreshkin et al., 2020) fail to properly extrapolate the pendulum dynamics. Incomplete physical models (Hamiltonian and ParamODE (ω_0)) are even worse since they do not account for friction. Augmenting them with APHYNITY significantly and consistently improves forecasting results and parameter identification.

Table 9: Forecasting and identification results on the damped pendulum dataset with different parameters for each sequence. log MSEs are computed over 20 predicted time-steps. For each level of incorporated physical knowledge, equivalent best results according to a Student t-test are shown in bold. n/a corresponds to non-applicable cases.

	Method	log MSE	%Error T_0	%Error α	$\|F_a\|^2$
data-driven	Neural ODE (Chen et al., 2018)	-4.35±0.9	n/a	n/a	n/a
	N-Beats (Oreshkin et al., 2020)	-4.57±0.5	n/a	n/a	n/a
Incomplete physics	Hamiltonian (Greydanus et al., 2019)	-1.31±0.4	n/a	n/a	n/a
	APHYNITY Hamiltonian	**-4.72±0.4**	n/a	n/a	5.6±0.6
	Param ODE (ω_0)	-2.66±0.9	21.5±19	n/a	n/a
	APHYNITY Param ODE (ω_0)	**-5.94±0.7**	**5.0±1.8**	n/a	0.49±0.1
Complete physics	Param ODE (ω_0, α)	**-5.71±0.4**	4.08±0.8	152±129	n/a
	APHYNITY Param ODE (ω_0, α)	**-6.22±0.7**	**3.26±0.6**	**62±27**	(5.39±0.1)e-10
	True ODE	**-8.58±0.1**	n/a	n/a	n/a
	APHYNITY True ODE	**-8.58±0.1**	n/a	n/a	(2.15±1.6)e-4

H EXTENSION TO A PARTIALLY-OBSERVED SETTING: VIDEO PREDICTION

We investigate here the applicability of APHYNITY in a partially-observed context. We consider the Moving MNIST video prediction problem, which is a standard benchmark (Srivastava et al., 2015) consisting in 2 MNIST digits moving with constant translation and bouncing on the walls. For this problem, we use the APHYNITY framework on top of the recent PhyDNet architecture (Le Guen & Thome, 2020) that applies PDE dynamics and residual dynamics in a learned latent space, but without any cooperation scheme. The challenge here is to jointly learn an adequate latent representation of the state directly from pixels and the correct decomposition between the physical and augmentation components.

As done classically (Srivastava et al., 2015), we predict 10 future frames given 10 input frames. Training sequences (64×64 grayscale images) are generated on the fly and the fixed test set of 5000 sequences is provided by (Srivastava et al., 2015).

Table 10: Video prediction results on Moving MNIST.

Method	MSE (↓)	MAE (↓)	SSIM (↑)
ConvLSTM (Shi et al., 2015)	103.3	182.9	0.707
PredRNN (Wang et al., 2017)	56.8	126.1	0.867
PredRNN++ (Wang et al., 2018)	46.5	106.8	0.898
MIM (Wang et al., 2019c)	44.2	101.1	0.910
E3D-LSTM (Wang et al., 2019b)	41.3	86.4	0.920
DDPAE (Hsieh et al., 2018)	38.9	90.7	0.922
PhyDNet (Le Guen & Thome, 2020)	24.4	70.3	0.947
APHYNITY **(ours)**	**22.8**	**67.2**	**0.950**

Video prediction results are reported in Table 10, evaluated with mean squared error (MSE), mean absolute error (MSE) and structural similarity (SSIM). We see that APHYNITY reaches new state-of-the-art performances for this dataset, outperforming in all metrics recent competitive data-driven baselines (Shi et al., 2015; Wang et al., 2017; 2019b;c;b; Hsieh et al., 2018). Note also that APHYNITY outperforms PhyDNet (Le Guen & Thome, 2020) in forecasting performances, confirming that the proper MB/ML decomposition ensured by APHYNITY is fundamental for final performances.