Generalizing to New Physical Systems via Context-Informed Dynamics Model

Matthieu Kirchmeyer, Yuan Yin, Jérémie Donà, Nicolas Baskiotis, Alain Rakotomamonjy, Patrick Gallinari

Generalizing to New Physical Systems via Context-Informed Dynamics Model

Matthieu Kirchmeyer [1 2 *] **Yuan Yin** [1 *]
Jérémie Donà [1] **Nicolas Baskiotis** [1] **Alain Rakotomamonjy** [2 3] **Patrick Gallinari** [1 2]

Abstract

Data-driven approaches to modeling physical systems fail to generalize to unseen systems that share the same general dynamics with the learning domain, but correspond to different physical contexts. We propose a new framework for this key problem, context-informed dynamics adaptation (CoDA), which takes into account the distributional shift across systems for fast and efficient adaptation to new dynamics. CoDA leverages multiple environments, each associated to a different dynamic, and learns to condition the dynamics model on contextual parameters, specific to each environment. The conditioning is performed via a hypernetwork, learned jointly with a context vector from observed data. The proposed formulation constrains the search hypothesis space for fast adaptation and better generalization across environments with few samples. We theoretically motivate our approach and show state-of-the-art generalization results on a set of nonlinear dynamics, representative of a variety of application domains. We also show, on these systems, that new system parameters can be inferred from context vectors with minimal supervision.

1. Introduction

Neural Network (NN) approaches to modeling dynamical systems have recently raised the interest of several communities leading to an increasing number of contributions. This topic was explored in several domains, ranging from simple dynamics e.g. Hamiltonian systems (Greydanus et al., 2019; Chen et al., 2020b) to more complex settings e.g. fluid dynamics (Kochkov et al., 2021; Li et al., 2021; Wandel et al., 2021), earth system science and climate (Reichstein et al., 2019), or health (Fresca et al., 2020). NN emulators are attractive as they may for example provide fast and low cost approximations to complex numerical simulations (Duraisamy et al., 2019; Kochkov et al., 2021), complement existing simulation models when the physical law is partially known (Yin et al., 2021b) or even offer solutions when classical solvers fail e.g. with very high number of variables (Sirignano & Spiliopoulos, 2018).

A model of a real-world dynamical system should account for a wide range of contexts resulting from different external forces, spatio-temporal conditions, boundary conditions, sensors characteristics or system parameters. These contexts characterize the dynamics phenomenon. For instance, in cardiac electrophysiology (Neic et al., 2017; Fresca et al., 2020), each patient has its own specificities and represents a particular context. In the study of epidemics' diffusion (Shaier et al., 2021), computational models should handle a variety of spatial, temporal or even sociological contexts. The same holds for most physical problems, e.g. forecasting of spatial-location-dependent dynamics in climate (de Bézenac et al., 2018), fluid dynamics prediction under distinct external forces (Li et al., 2021), etc.

The physics approach for modeling dynamical systems relies on a strong prior knowledge about the underlying phenomenon. This provides a causal mechanism which is embedded in a physical dynamics model, usually a system of differential equations, and allows the physical model to handle a whole set of contexts. Moreover, it is often possible to adapt the model to new or evolving situations, e.g. via data assimilation (Kalman, 1960; Courtier et al., 1994).

In contrast, Expected Risk Minimization (ERM) based machine learning (ML) fails to generalize to unseen dynamics. Indeed, it requires i.i.d. data for training and inference while dynamical observations are non-i.i.d. as the distributions change with initial conditions or physical contexts.

Thus any ML framework that handles this question should consider other assumptions. A common one used e.g. in domain generalization (Wang et al., 2021b), states that data come from several environments a.k.a. domains, each with a different distribution. Training is performed on a sample of the environments and test corresponds to new ones. Do-

[*]Equal contribution [1]CNRS-ISIR, Sorbonne University, Paris, France [2]Criteo AI Lab, Paris, France [3]Université de Rouen, LITIS, France. Correspondence to: Matthieu Kirchmeyer <matthieu.kirchmeyer@sorbonne-universite.fr>, Yuan Yin <yuan.yin@sorbonne-universite.fr>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

main generalization methods attempt to capture problem invariants via a unique model, assuming that there exists a representation space suitable for all the environments. This might be appropriate for classification, but not for dynamical systems where the underlying dynamics differs for each environment. For this problem, we need to learn a function that adapts to each environment, based on a few observations, instead of learning a single domain-invariant function. This is the objective of meta-learning (Thrun & Pratt, 1998), a general framework for fast adaptation to unknown contexts. The standard gradient-based methods (e.g. Finn et al., 2017) are unsuitable for complex dynamics due to their bi-level optimization and are known to overfit when little data is available for adaptation, as in the few-shot learning setting explored in this paper (Mishra et al., 2018). Like invariant methods, meta-learning usually handles basic tasks e.g. classification; regression on static data or simple sequences and not challenging dynamical systems.

Generalization for modeling real-world dynamical systems is a recent topic. Simple simulated dynamics were considered in Reinforcement Learning (Lee et al., 2020; Clavera et al., 2019) while physical dynamics were modeled in recent works (Yin et al., 2021a; Wang et al., 2021c). These approaches consider either simplified settings or additional hypotheses e.g. prior knowledge and do not offer general solutions to our adaptation problem (details in Section 6).

We propose a new ML framework for generalization in dynamical systems, called **Co**ntext-Informed **D**ynamics **A**daptation (CoDA). Like in domain generalization, we assume availability of several environments, each with its own specificity, yet sharing some physical properties. Training is performed on a sample of the environments. At test time, we assume access to example data from a new environment, here a trajectory. Our goal is to adapt to the new environment distribution with this trajectory. More precisely, CoDA assumes that the underlying system is described by a parametrized differential equation, either an ODE or a PDE. The environments share the parametrized form of the equation but differ by the values of the parameters or initial conditions. CoDA conditions the dynamics model on learned environment characteristics a.k.a. contexts and generalizes to new environments and trajectories with few data. Our main contributions are the following:

- We introduce a multi-environment formulation of the generalization problem for dynamical systems.
- We propose a novel context-informed framework, CoDA, to this problem. It conditions the dynamics model on context vectors via a hypernetwork. CoDA introduces a locality and a low-rank constraint, which enable fast and efficient adaptation with few data.
- We analyze theoretically the validity of our low-rank adaptation setting for modeling dynamical systems.
- We evaluate two variations of CoDA on several ODEs/PDEs representative of a variety of application domains, e.g. chemistry, biology, physics. CoDA achieves SOTA generalization results on in-domain and one-shot adaptation scenarios. We also illustrate how, with minimal supervision, CoDA infers accurately new system parameters from learned contexts.

The paper is organized as follows. In Section 2, we present our multi-environment problem. In Section 3, we introduce the CoDA framework. In Section 4, we detail how to implement our framework. In Section 5, we present our experimental results. In Section 6, we present related work.

2. Generalization for Dynamical Systems

We present our generalization problem for dynamical systems, then introduce our multi-environment formalization.

2.1. Problem Setting

We consider dynamical systems that are driven by unknown temporal differential equations of the form:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = f(x(t)), \tag{1}$$

where $t \in \mathbb{R}$ is a time index, $x(t)$ is a time-dependent state in a space $\mathcal{X}$ and $f : \mathcal{X} \to T\mathcal{X}$ a function that maps $x(t) \in \mathcal{X}$ to its temporal derivatives in the tangent space $T\mathcal{X}$. f belongs to a class of vector fields $\mathcal{F}$. $\mathcal{X} \subseteq \mathbb{R}^d$ ($d \in \mathbb{N}^\star$) for ODEs or $\mathcal{X}$ is a space of functions defined over a spatial domain (e.g. 2D or 3D Euclidean space) for PDEs.

Functions $f \in \mathcal{F}$ define a space $\mathcal{D}^f(\mathcal{X})$ of state trajectories $x : I \to \mathcal{X}$, mapping t in an interval I including 0, to the state $x(t) \in \mathcal{X}$. Trajectories are defined by the initial condition $x(0) \triangleq x_0 \sim p(X_0)$ and take the form:

$$\forall t \in I, x(t) = x_0 + \int_0^t f(x(\tau))\mathrm{d}\tau \in \mathcal{X} \tag{2}$$

In the following, we assume that $f \in \mathcal{F}$ is parametrized by some unknown attributes e.g. physical parameters, external forcing terms which affect the trajectories.

2.2. Multi-Environment Learning Problem

We propose to learn the class of functions $\mathcal{F}$ with a data-driven *dynamics model* g_θ parametrized by $\theta \in \mathbb{R}^{d_\theta}$. Given $f \in \mathcal{F}$, we observe N trajectories in $\mathcal{D}^f(\mathcal{X})$ (cf. Eq. (2)).

The standard ERM objective considers that all trajectories are i.i.d. Here, we propose a multi-environment learning formulation where observed trajectories of f form an environment $e \in \mathcal{E}$. We denote f^e and $\mathcal{D}^e$ the corresponding function and set of N trajectories. We assume that we

observe training environments $\mathcal{E}_{\text{tr}}$, consisting of several trajectories from a set of known functions $\{f^e\}_{e\in\mathcal{E}_{\text{tr}}}$.

The goal is to learn g_θ that adapts easily and efficiently to new environments $\mathcal{E}_{\text{ad}}$, corresponding to unseen functions $\{f^e\}_{e\in\mathcal{E}_{\text{ad}}}$ ("ad" stands for adaptation). We define $\forall e \in \mathcal{E}$ the Mean Squared Error (MSE) loss, over $\mathcal{D}^e$ as

$$\mathcal{L}(\theta, \mathcal{D}^e) \triangleq \sum_{i=1}^{N} \int_{t\in I} \|f^e(x^{e,i}(t)) - g_\theta(x^{e,i}(t))\|_2^2 \mathrm{d}t \quad (3)$$

In practice, f^e is unavailable and we can only approximate it from discretized trajectories. We detail later in Eq. (10) our approximation method based on an integral formulation. It fits observed trajectories directly in state space.

3. The CoDA Learning Framework

We introduce CoDA, a new context-informed framework for learning dynamics on multiple environments. It relies on a general adaptation rule (Section 3.1) and introduces two key properties: locality, enforced in the objective (Section 3.2) and low-rank adaptation, enforced in the proposed model via hypernetwork-decoding (Section 3.3). The validity of this framework for dynamical systems is analyzed in Section 3.4 and its benefits are discussed in Section 3.5.

3.1. Adaptation Rule

The dynamics model g_θ should adapt to new environments. Hence, we propose to condition g_θ on observed trajectories $\mathcal{D}^e, \forall e \in \mathcal{E}$. Conditioning is performed via an *adaptation network* A_π, parametrized by π, which adapts the weights of g_θ to an environment $e \in \mathcal{E}$ according to

$$\theta^e \triangleq A_\pi(\mathcal{D}^e) \triangleq \theta^c + \delta\theta^e, \quad \pi \triangleq \{\theta^c, \{\delta\theta^e\}_{e\in\mathcal{E}}\} \quad (4)$$

$\theta^c \in \mathbb{R}^{d_\theta}$ are shared parameters, used as an initial value for fast adaptation to new environments. $\delta\theta^e \in \mathbb{R}^{d_\theta}$ are environment-specific parameters conditioned on $\mathcal{D}^e$.

3.2. Constrained Optimization Problem

Given the adaptation rule in Eq. (4), we introduce a constrained optimization problem which learns parameters π such that $\forall e \in \mathcal{E}, \delta\theta^e$ is small and g fits observed trajectories. It introduces a locality constraint with a norm $\|\cdot\|$:

$$\min_\pi \sum_{e\in\mathcal{E}} \|\delta\theta^e\|^2 \text{ s.t. } \forall x^e(t) \in \mathcal{D}^e, \frac{\mathrm{d}x^e(t)}{\mathrm{d}t} = g_{\theta^c+\delta\theta^e}(x^e(t))$$

We consider an approximation of this problem which relaxes the equality constraint with the MSE loss $\mathcal{L}$ in Eq. (3).

$$\min_\pi \sum_{e\in\mathcal{E}} \left(\mathcal{L}(\theta^c + \delta\theta^e, \mathcal{D}^e) + \lambda\|\delta\theta^e\|^2\right) \quad (5)$$

λ is a hyperparameter. For training, we minimize Eq. (5) w.r.t. π over training environments $\mathcal{E}_{\text{tr}}$. After training, θ^c is freezed. For adaptation, we minimize Eq. (5) over new environments $\mathcal{E}_{\text{ad}}$ w.r.t. $\{\delta\theta^e\}_{e\in\mathcal{E}_{\text{ad}}}$.

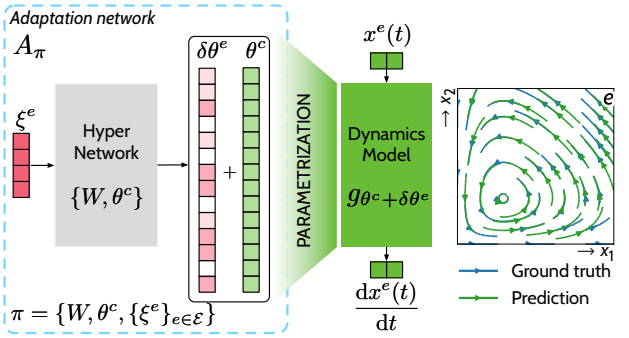


Figure 1. Context-Informed Dynamics Adaptation (CoDA).

The locality constraint in the training objective Eq. (5) enforces $\delta\theta^e$ to remain close to the shared θ^c solutions. It plays several roles. First, it fosters fast adaptation by acting as a constraint over $\theta^c \in \mathbb{R}^{d_\theta}$ during training s.t. minimas $\{\theta^{e\star}\}_{e\in\mathcal{E}}$ are in a neighborhood of θ^c i.e. can be reached from θ^c with few update steps. Second, it constrains the hypothesis space at fixed θ^c. Under some assumptions, it can simplify the resolution of the optimization problem w.r.t. $\delta\theta^e$ by turning optimization to a quadratic convex problem with an unique solution. We show this property for our solution in Proposition 1. The positive effects of this constraint will be illustrated on an ODE system in Section 3.3.

3.3. Context-Informed Hypernetwork

Eq. (5) involves learning $\delta\theta^e$ for each environment. For adaptation, $\delta\theta^e$ should be inferred from few observations of the new environment. Learning such high-dimensional parameters is prone to over-fitting, especially in low data regimes. We propose a hypernetwork-based solution (Figure 1) to solve efficiently this problem. It operates on a low-dimensional space, yields fixed-cost adaptation and shares efficiently information across environments.

Formulation We estimate $\delta\theta^e$ through a linear mapping of conditioning information, called context, learned from $\mathcal{D}^e$ and denoted $\xi^e \in \mathbb{R}^{d_\xi}$. $W = (W_1, \cdots, W_{d_\xi}) \in \mathbb{R}^{d_\theta \times d_\xi}$ is the weight matrix of the linear decoder s.t.

$$A_\pi(\mathcal{D}^e) \triangleq \theta^c + W\xi^e, \quad \pi \triangleq \{W, \theta^c, \{\xi^e\}_{e\in\mathcal{E}}\} \quad (6)$$

W is shared across environments and defines a low-dimensional subspace $\mathcal{W} \triangleq \text{Span}(W_1, ..., W_{d_\xi})$, of dimension at most d_ξ, to which the search space of $\delta\theta^e$ is restricted. ξ^e is specific to each environment and can be interpreted as learning rates along the rows of W. In our experiments, $d_\xi \ll d_\theta$ is small, at most 2. Thus, *adaptation to new environments only requires to learn very few parameters, which define a completely new dynamics model g.*

A_π corresponds to an affine mapping of ξ^e parametrized by $\{W, \theta^c\}$, a.k.a. a linear hypernetwork. Note that hypernetworks (Ha et al., 2017) have been designed to handle single-environment problems and learn a separate context per layer. Our formalism involves multiple environments and defines a context per environment for all layers of g.

Linearity of the hypernetwork is not restrictive as contexts are directly learned through an inverse problem detailed in eqs. (7) and (8), s.t. expressivity is similar to a nonlinear hypernetwork with a final linear activation.

Objectives We derive the training and adaptation objectives by inserting Eq. (6) into Eq. (5). For training, both contexts and hypernetwork are learned with Eq. (7):

$$\min_{\theta^c, W, \{\xi^e\}_{e\in\mathcal{E}_{\text{tr}}}} \sum_{e\in\mathcal{E}_{\text{tr}}} \Big(\mathcal{L}(\theta^c + W\xi^e, \mathcal{D}^e) + \lambda\|W\xi^e\|^2\Big) \quad (7)$$

After training, θ^c is kept fixed and for adaptation to a new environment, only the context vector ξ^e is learned with:

$$\min_{\{\xi^e\}_{e\in\mathcal{E}_{\text{ad}}}} \sum_{e\in\mathcal{E}_{\text{ad}}} \Big(\mathcal{L}(\theta^c + W\xi^e, \mathcal{D}^e) + \lambda\|W\xi^e\|^2\Big) \quad (8)$$

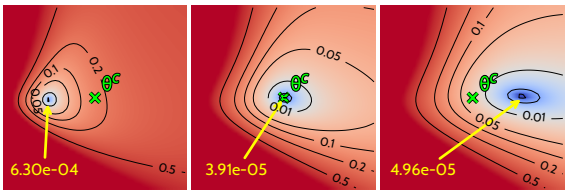
Implementation of eqs. (7) and (8) is detailed in Section 4. We apply gradient descent. In Proposition 1, we show for $\|\cdot\| = \ell_2$, that Eq. (8) admits an unique solution, recovered from initialization at $\mathbf{0}$ with a single preconditioned gradient step, projected onto subspace $\mathcal{W}$ defined by W.

Proposition 1 (Proof in Appendix B)**.** *Given $\{\theta^c, W\}$ fixed, if $\|\cdot\| = \ell_2$, then Eq. (8) is quadratic. If $\lambda' W^\top W$ or $\bar{H}^e(\theta^c) = W^\top \nabla^2_\theta \mathcal{L}(\theta^c, \mathcal{D}^e) W$ are invertible then $\bar{H}^e(\theta^c) + \lambda' W^\top W$ is invertible except for a finite number of λ' values. The problem in Eq. (8) is then also convex and admits an unique solution, $\{\xi^{e\star}\}_{e\in\mathcal{E}_{ad}}$. With $\lambda' \triangleq 2\lambda$,*

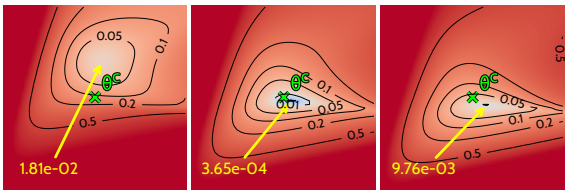
$$\xi^{e*} = -\Big(\bar{H}^e(\theta^c) + \lambda' W^\top W\Big)^{-1} W^\top \nabla_\theta \mathcal{L}(\theta^c, \mathcal{D}^e) \quad (9)$$

Interpretation We now interpret CoDA by visualizing its loss landscape in Figure 2a and comparing it to ERM's loss landscape in Figure 2b. We use the package in Li et al. (2018b) to plot loss landscapes around θ^c and consider the *Lotka-Volterra* system, described in Section 5.1.

In Figure 2a, loss values of CoDA are projected onto subspace $\mathcal{W}$, where $d_\xi = 2$. We make three observations. First, across environments, the loss is smooth and has a single minimum around θ^c. Second, the local optimum of the loss is close to θ^c across environments. Finally, the minimal loss value on $\mathcal{W}$ around θ^c is low across environments. The two first properties were discussed in Section 3.2 and are a direct consequence of the locality constraint on $\mathcal{W}$. When $\|\cdot\| = \ell_2$, it makes the optimization problem in Eq. (7) quadratic w.r.t. ξ^e and convex under invertibility of



(a) CoDA's loss projected onto $\mathcal{W} = \text{Span}(W_1, W_2)$.



(b) ERM's loss projected onto the span of the two principal gradient directions computed with SVD.

Figure 2. Loss landscapes centered in θ^c, marked with ×, for 3 environments on the *Lotka-Volterra* ODE. $\forall e$, → points to the local optimum $\theta^{e\star}$ with loss value reported in yellow.

$\bar{H}^e(\theta^c) + \lambda' W^\top W$ as detailed in Proposition 1. We provided in Eq. (9) the closed form expression of the solution. It also imposes small $\|\xi^e\|$ s.t. when minimizing the loss in Eq. (7), θ^c remains close to local optimas of all training environments. The final observation illustrates that CoDA finds a subspace $\mathcal{W}$ with environment-specific parameters of low loss values i.e. low-rank adaptation performs well.

In Figure 2b, loss values of ERM are projected onto the span of the two principal gradient directions. We observe that, unlike CoDA, ERM does not find low loss values. Indeed, it aims at finding θ^c with good performance across environments, thus cannot model several dynamics.

3.4. Validity for Dynamical Systems

We further motivate low-rank decoding in our context-informed hypernetwork approach by providing some evidence that gradients at θ^c across environments define a low-dimensional subspace. We consider the loss $\mathcal{L}$ in Eq. (3) and define the gradient subspace in Definition 1.

Definition 1 (Gradient directions)**.** *With $\mathcal{L}$ in Eq. (3), $\forall \theta^c \in \mathbb{R}^{d_\theta}$ parametrizing a dynamics model g_{θ^c}, the subspace generated by gradient directions at θ^c across environments $\mathcal{E}$ is denoted $\mathcal{G}_{\theta^c} \triangleq \text{Span}(\{\nabla_\theta \mathcal{L}(\theta^c, \mathcal{D}^e)\}_{e\in\mathcal{E}})$.*

We show, in Proposition 2, low-dimensionality of $\mathcal{G}_{\theta^c}$ for linearly parametrized systems.

Proposition 2 (Low-rank under linearity. Proof in Appendix B)**.** *Given a class of linearly parametrized dynamics $\mathcal{F}$ with d_p varying parameters, $\forall \theta^c \in \mathbb{R}^{d_\theta}$, subspace $\mathcal{G}_{\theta^c}$ in Definition 1 is low-dimensional and $\dim(\mathcal{G}_{\theta^c}) \leq d_p \ll d_\theta$.*

The linearity assumption is not restrictive as it is present in a wide variety of real-world systems e.g. Burger or Ko-

rteweg–De Vries PDE (Raissi et al., 2019), convection-diffusion (Long et al., 2018), wave and reaction diffusion equations (Yin et al., 2021b) etc.

Under nonlinearity, we do not have the same theoretical guarantee, yet, we show empirically in Appendix D that low-dimensionality of parameters of the dynamics model still holds for several systems. This property is comforted by recent work that highlighted that gradients are low-rank throughout optimization in single-domain settings, i.e. that the solution space is low-dimensional (Gur-Ari et al., 2019; Li et al., 2018a;b). In the same spirit as CoDA, this property was leveraged to design efficient solutions to the learning problems (Frankle & Carbin, 2019; Vogels et al., 2019).

3.5. Benefits of CoDA

We highlight the benefits of CoDA. CoDA is a general time-continuous framework that can be used with any approximator g_θ of the derivative Eq. (3). It can be trained with a given temporal resolution and tested on another; it handles irregularly-sampled sequences. The choice of the approximator g_θ defines the ability to handle different spatial resolutions for PDEs, as further detailed in Section 5.3.

Compared to related adaptation methods, CoDA presents several advantages. First, as detailed in Appendix A.1, the adaptation rule in Eq. (4) is similar to the one used in gradient-based meta-learning; yet, our first order joint optimization problem in Eq. (5) simplifies the complex bi-level optimization problem (Antoniou et al., 2019). Second, CoDA introduces the two key properties of locality constraint and low-rank adaptation which guarantee efficient adaptation to new environments as discussed in Section 3.3. Third, it generalizes contextual meta-learning methods (Garnelo et al., 2018; Zintgraf et al., 2019), which also perform low-rank adaptation, via the hypernetwork decoder (details in Appendix A.2). Our decoder learns complex environment-conditional dynamics models while controlling their complexity. Finally, CoDA learns context vectors through an inverse problem as Zintgraf et al. (2019). This decoder-only strategy is particularly efficient and flexible in our setting. An alternative is to infer them via a learned encoder of $\mathcal{D}^e$ as Garnelo et al. (2018). Yet, the latter was observed to underfit (Kim et al., 2019), requiring extensive tuning of the encoder and decoder architecture. Overall, CoDA is easy to implement and maintains expressivity with a linear decoder.

4. Framework Implementation

We detail how to perform trajectory-based learning with our framework and describe two instantiations of the locality constraint. We detail the corresponding pseudo-code.

Trajectory-Based Formulation As derivatives in Eq. (3) are not directly observed, we use in practice for training a trajectory-based formulation of Eq. (3). We consider a set of N trajectories, $\mathcal{D}^e$. Each trajectory is discretized over a uniform temporal and spatial grid and includes $\frac{T}{\Delta t}\left(\frac{S}{\Delta s}\right)^{d_s}$ states, where d_s is the spatial dimension for PDEs and $d_s = 0$ for ODEs. $\Delta t, \Delta s$ are the temporal and spatial resolutions and T, S the temporal horizon and spatial grid size.

Our loss writes as:

$$\mathcal{L}(\theta, \mathcal{D}^e) = \sum_{i=1}^{N} \sum_{j=1}^{(S/\Delta s)^{d_s}} \sum_{k=1}^{T/\Delta t} \left\| x^{e,i}(t_k, s_j) - \tilde{x}^{e,i}(t_k, s_j) \right\|_2^2$$

$$\text{where } \tilde{x}^{e,i}(t_k) = x_0^{e,i} + \int_{t_0}^{t_k} g_\theta\big(\tilde{x}^{e,i}(\tau)\big) \mathrm{d}\tau \tag{10}$$

$x^{e,i}(t_k, s_j)$ is the state value in the i^{th} trajectory from environment e at the j^{th} spatial coordinate s_j and time $t_k \triangleq k\Delta t$. $x^{e,i}(t) \triangleq [x(t, s_1), \cdots, x(t, s_{(S/\Delta s)^{d_s}})]^\top$ is the state vector in the i^{th} trajectory from environment e over the spatial domain at time t and $x_0^{e,i}$ is the corresponding initial condition. To compute $\tilde{x}^{e,i}(t_k)$, we apply for integration a numerical solver (Hairer et al., 2000) as detailed later.

Locality Constraint Instead of penalizing $\lambda \|W\xi^e\|^2$ in Eq. (7), we found it more efficient to penalize separately W and ξ^e. We thus introduce the following regularization:

$$R(W, \xi^e) \triangleq \lambda_\xi \|\xi^e\|_2^2 + \lambda_\Omega \Omega(W) \tag{11}$$

It involves hyperparameters $\lambda_\xi, \lambda_\Omega$ and a norm $\Omega(W)$ which depends on the choice of $\|\cdot\|$ in Eq. (5). Minimizing $R(W, \xi^e)$ minimizes an upper-bound to $\|\cdot\|$, derived in appendix E for the two considered variations of $\|\cdot\|$:

- CoDA-ℓ_2 sets $\|\cdot\| \triangleq \ell_2(\cdot)$ and $\Omega \triangleq \ell_2^2$, constraining $W\xi^e$ to a sphere.
- CoDA-ℓ_1 sets $\|\cdot\| \triangleq \ell_1(\cdot)$ and $\Omega = \ell_{1,2}$ over rows i.e. $\Omega(W) \triangleq \sum_{i=1}^{d_\theta} \|W_{i,:}\|_2$ to induce sparsity and find most important parameters for adaptation. $\ell_{1,2}$ constrains $\mathcal{W}$ to be axis-aligned; then the number of solutions is finite as $\dim(\mathcal{W})$ is finite.

Pseudo-Code We solve Eq. (7) for training and Eq. (8) for adaptation using eqs. (10) and (11) and Algorithm 1. We back-propagate through the solver with torchdiffeq (Chen, 2021) and apply exponential Scheduled Sampling (Goyal et al., 2016) to stabilize training. We provide our code at `https://github.com/yuan-yin/CoDA`.

5. Experiments

We validate our approach on four classes of challenging nonlinear temporal and spatiotemporal physical dynamics,

Table 1. Test MSE (↓) in training environments $\mathcal{E}_{\text{tr}}$ (*In-Domain*), new environments $\mathcal{E}_{\text{ad}}$ (*Adaptation*). Best in **bold**; second underlined.

	LV ($\times 10^{-5}$)		GO ($\times 10^{-4}$)		GS ($\times 10^{-3}$)		NS ($\times 10^{-4}$)	
	IN-DOMAIN	ADAPTATION	IN-DOMAIN	ADAPTATION	IN-DOMAIN	ADAPTATION	IN-DOMAIN	ADAPTATION
MAML	60.3±1.3	3150±940	57.3±2.1	1081±62	3.67±0.53	2.25±0.39	68.0±8.0	51.1±4.0
ANIL	381±76	4570±2390	74.5±11.5	1688±226	5.01±0.80	3.95±0.11	61.7±4.3	48.6±3.2
META-SGD	32.7±12.6	7220±4580	42.3±6.9	1573±413	2.85±0.54	2.68±0.20	53.9±28.1	44.3±27.1
LEADS	3.70±0.27	47.61±12.47	31.4±3.3	113.8±41.5	2.90±0.76	1.36±0.43	14.0±1.55	28.6±7.23
CAVIA-FILM	4.38±1.15	8.41±3.20	4.44±1.46	3.87±1.28	2.81±1.15	1.43±1.07	23.2±12.1	22.6±9.88
CAVIA-CONCAT	2.43±0.66	6.26±0.77	5.09±0.35	2.37±0.23	2.67±0.48	1.62±0.85	25.5±6.31	26.0±8.24
CODA-ℓ_2	1.52±0.08	1.82±0.24	2.45±0.38	1.98±0.06	1.01±0.15	0.77±0.10	9.40±1.13	10.3±1.48
CODA-ℓ_1	**1.35**±0.22	**1.24**±0.20	**2.20**±0.26	**1.86**±0.29	**0.90**±0.057	**0.74**±0.10	**8.35**±1.71	**9.65**±1.37

Algorithm 1 CoDA Pseudo-code

Training:

Input: $\mathcal{E}_{\text{tr}} \subset \mathcal{E}$, $\{\mathcal{D}^{e_{\text{tr}}}\}_{e_{\text{tr}} \in \mathcal{E}_{\text{tr}}}$ with $\forall e_{\text{tr}} \in \mathcal{E}_{\text{tr}}, \#\mathcal{D}^{e_{\text{tr}}} = N_{\text{tr}}$;
$\pi = \{W, \theta^c, \{\xi^{e_{\text{tr}}}\}_{e_{\text{tr}} \in \mathcal{E}_{\text{tr}}}\}$ where $W \in \mathbb{R}^{d_\theta \times d_\xi}, \theta^c \in \mathbb{R}^{d_\theta}$ randomly initialized and $\forall e_{\text{tr}} \in \mathcal{E}_{\text{tr}}, \xi^{e_{\text{tr}}} = \mathbf{0} \in \mathbb{R}^{d_\xi}$.
loop
$\pi \leftarrow \pi - \eta \nabla_\pi \Big(\sum_{e_{\text{tr}} \in \mathcal{E}_{\text{tr}}} \mathcal{L}(\theta^c + W\xi^{e_{\text{tr}}}, \mathcal{D}^{e_{\text{tr}}}) + R(W, \xi^{e_{\text{tr}}}) \Big)$

Adaptation:

Input: $e_{\text{ad}} \in \mathcal{E}_{\text{ad}}$; $\mathcal{D}^{e_{\text{ad}}}$ with $\#\mathcal{D}^{e_{\text{ad}}} = N_{\text{ad}}$;
Trained $W \in \mathbb{R}^{d_\theta \times d_\xi}, \theta^c \in \mathbb{R}^{d_\theta}$ and $\xi^{e_{\text{ad}}} = \mathbf{0} \in \mathbb{R}^{d_\xi}$.
loop
$\xi^{e_{\text{ad}}} \leftarrow \xi^{e_{\text{ad}}} - \eta \nabla_{\xi^{e_{\text{ad}}}} \Big(\mathcal{L}(\theta^c + W\xi^{e_{\text{ad}}}, \mathcal{D}^{e_{\text{ad}}}) + R(W, \xi^{e_{\text{ad}}}) \Big)$

representative of various fields e.g. chemistry, biology and fluid dynamics. We evaluate in-domain and adaptation prediction performance and compare them to related baselines. We also investigate how learned context vectors can be used for system parameter estimation. We consider a few-shot adaptation setting where only few trajectories (N_{ad}) are available at adaptation time on new environments.

5.1. Dynamical Systems

We consider four ODEs and PDEs described in Appendix F.1. ODEs include *Lotka-Volterra* (LV, Lotka, 1925) and *Glycolitic-Oscillator* (GO, Daniels & Nemenman, 2015), modelling respectively predator-prey interactions and the dynamics of yeast glycolysis. PDEs are defined over a 2D spatial domain and include *Gray-Scott* (GS, Pearson, 1993), a reaction-diffusion system with complex spatiotemporal patterns and the challenging *Navier-Stokes* system (NS, Stokes, 1851) for incompressible flows. All systems are nonlinear w.r.t. system states and all but GO are linearly parametrized. The analysis in Section 3.4 covers all systems but GO. Experiments on the latter show that CoDA also extends to nonlinearly parametrized systems.

5.2. Experimental Setting

We consider forecasting: only the initial condition is used for prediction. We perform two types of evaluation: in-domain generalization on $\mathcal{E}_{\text{tr}}$ *(In-domain)* and out-of-domain adaptation to new environments $\mathcal{E}_{\text{ad}}$ (*Adaptation*). Each environment $e \in \mathcal{E}$ is defined by system parameters and $p^e \in \mathbb{R}^{d_p}$ denotes those that vary across $\mathcal{E}$. d_p represents the degrees of variations in $\mathcal{F}$; $d_p = 2$ for LV, GO, GS and $d_p = 1$ for NS. Appendix F.1 defines for each system the number of training and adaptation environments ($\#\mathcal{E}_{\text{tr}}$ and $\#\mathcal{E}_{\text{ad}}$) and the corresponding parameters. Appendix F.1 also reports the number of trajectories N_{tr} per training environment in $\mathcal{E}_{\text{tr}}$ and the distribution $p(X_0)$ from which are sampled all initial conditions (including adaptation and evaluation initial conditions). For *Adaptation*, we consider $N_{\text{ad}} = 1$ trajectory per new environment in $\mathcal{E}_{\text{ad}}$ to infer the context vector with Eq. (8). We consider more trajectories per adaptation environment in Section 5.7.

Evaluation is performed on 32 new test trajectories per environment. We report, in our tables, mean and standard deviation of Mean Squared Error (MSE) across test trajectories (Eq. (10)) over four different seeds. We report, in our figures, Mean Absolute Percentage Error (MAPE) in % over trajectories, as it allows to better compare performance across environments and systems. We define $\text{MAPE}(z, y)$ between a d-dimensional input z and target y as $\frac{1}{d} \sum_{j=1 \dots d : y_j \neq 0} \frac{|z_j - y_j|}{|y_j|}$. Over a trajectory, it extends into $\int_{t \in I} \text{MAPE}(\tilde{x}(t), x(t)) \mathrm{d}t$, with $\tilde{x}$ defined in Eq. (10).

5.3. Implementation of CoDA

We used for g_θ MLPs for ODEs, a resolution-dependent ConvNet for GS and a resolution-agnostic FNO (Li et al., 2021) for NS that can be used on new resolutions. Architecture details are provided in Appendix F.2. We tuned d_ξ and observed that $d_\xi = d_p$, the number of system parameters that vary across environments, performed best (cf. Section 5.6). We use Adam optimizer (Kingma & Ba, 2015) for all datasets; RK4 solver for LV, GS, GO and Euler solver for NS. Optimization and regularization hyperparameters are detailed in Appendix F.2.

5.4. Baselines

We consider three families of baselines, compared in Appendix Figure 6 and detailed in Section 6. First, Gradient-Based Meta-Learning (GBML) methods MAML (Finn

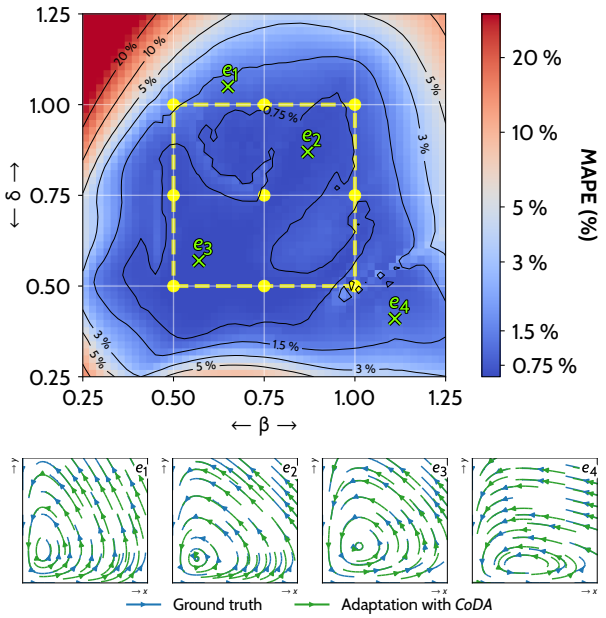


Figure 3. Adaptation results with CoDA-ℓ_1 on LV. Parameters (β, δ) are sampled in $[0.25, 1.25]^2$ on a 51×51 uniform grid, leading to 2601 adaptation environments $\mathcal{E}_{\text{ad}}$. • are training environments $\mathcal{E}_{\text{tr}}$. We report MAPE (↓) across $\mathcal{E}_{\text{ad}}$ (top). On the bottom, we choose four of them (×, e_1–e_4), to show the ground-truth (blue) and predicted (green) phase space portraits. x, y are respectively the quantity of prey and predator in the system in Eq. (15).

Table 2. Locality and *In-Domain* test MSE (↓). Best in **bold**.

	LV ($\times 10^{-5}$)		GO ($\times 10^{-4}$)	
CoDA	W/O ℓ_2	With ℓ_2	W/O ℓ_2	With ℓ_2
Full	2.28±0.29	1.52±0.08	2.98±0.71	2.45±0.38
FirstLayer	2.25±0.29	2.41±0.23	2.38±0.71	**2.12**±0.55
LastLayer	1.86±0.24	**1.27**±0.03	28.4±0.60	28.4±0.64

et al., 2017), ANIL (Rusu et al., 2019) and Meta-SGD (Li et al., 2017). Second, the Multi-Task Learning method LEADS (Yin et al., 2021a). Finally, the contextual meta-learning method CAVIA (Zintgraf et al., 2019), with conditioning via concatenation (Concat) or linear modulation of final hidden features (FiLM, Perez et al., 2018). All baselines are adapted to be dynamics-aware i.e. time-continuous: they consider the loss in Eq. (10), as CoDA. Moreover, they share the same architecture for g_θ as CoDA.

5.5. Generalization Results

In Table 1, we observe that CoDA improves significantly test MSE w.r.t. our baselines for both *In-Domain* and *Adaptation* settings. For PDE systems and a given test trajectory, we visualize in Figures 8 and 9 in Appendix G the predicted MSE by these models along the ground truth. We also notice improvements for CoDA over our baselines. Across datasets, all baselines are subject to a drop in performance

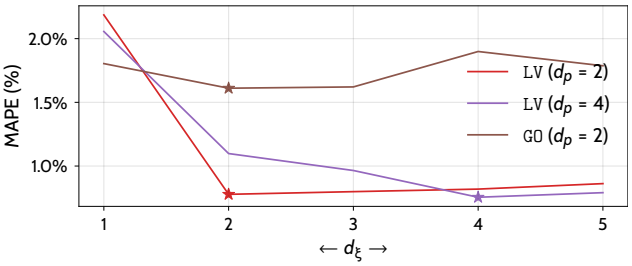


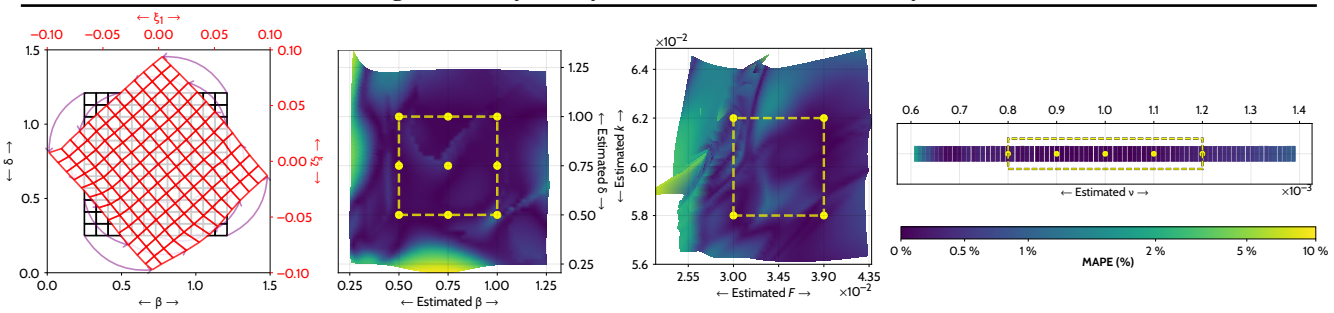
Figure 4. Dimension of the context vectors (d_ξ) and test *In-Domain* MAPE (↓) with CoDA-ℓ_1. "⋆" is the smallest MAPE.

between *In-Domain* and *Adaptation* while CoDA maintains remarkably the same level of performance in both cases. In more details, GBML methods (MAML, ANIL, Meta-SGD) overfit on training *In-Domain* data especially when data is scarce. This is the case for ODEs which include less system states for training than PDEs. LEADS performs better than GBML but overfits for *Adaptation* as it does not adapt efficiently. CAVIA-Concat/FiLM perform better than GBML and LEADS, as they leverage a context, but are less expressive than CoDA. Both variations of CoDA perform best as they combine the benefits of low-rank adaptation and locality constraint. CoDA-ℓ_1 is better than CoDA-ℓ_2 as it induces sparsity, further constraining the hypothesis space.

We evaluate in Figure 3 CoDA-ℓ_1 on LV for *Adaptation* over a wider range of adaptation environments ($\#\mathcal{E}_{\text{ad}} = 51 \times 51 = 2601$). We report mean MAPE over $\mathcal{E}_{\text{ad}}$ (top). We observe three regimes: inside the convex hull of training environments $\mathcal{E}_{\text{tr}}$, MAPE is very low; outside the convex-hull, MAPE remains low in a neighborhood of $\mathcal{E}_{\text{tr}}$; beyond this neighborhood, MAPE increases. CoDA thus generalizes efficiently in the neighborhood of training environments and degrades outside this neighborhood. We plot reconstructed phase space portraits (bottom) on four selected environments and observe that the learned solution (green) closely follows the target trajectories (blue).

5.6. Ablation Studies

We perform two studies on LV and GO. In a first study in Table 2, we evaluate the gains due to using ℓ_2 locality constraint on *In-Domain* evaluation. On line 1 (Full), we observe that CoDA-ℓ_2 performs better than CoDA without locality constraint. Prior work perform adaptation only on the final layer with some performance improvements on classification or Hamiltonian system modelling (Raghu et al., 2020; Chen et al., 2020a). In order to evaluate this strategy, we manually restrict hypernetwork-decoding to only one layer in the dynamics model g_θ, either the first layer (line 2) or the last layer (line 3). We observe that the importance of the layer depends on the parametrization of the system: for LV, linearly parametrized, the last layer is better while for GO, nonlinearly parametrized, the first layer is better. CoDA-ℓ_1 generalizes this idea by automatically se-



(a) LV: $(\beta, \delta) \in [0.25, 1.25]^2$ (b) GS: $F \in [2.25, 4.35] \cdot 10^{-2}$, $k \in [5.6, 6.4] \cdot 10^{-2}$ (c) NS: $\nu \in [0.6, 1.4] \cdot 10^{-3}$

Figure 5. Parameter estimation with CoDA-ℓ_1 in new adaptation environments on (a) LV, (b) GS and (c) NS. In (a), we visualize: on the left, context vectors ξ (red); on the right, true parameters (β, δ) (black). In (b) and (c), we visualize estimated parameters with corresponding estimation MAPE (↓). • are training environments $\mathcal{E}_{\text{tr}}$ with known parameters. - - delimits the convex hull of $\mathcal{E}_{\text{tr}}$.

Table 3. Test MSE $\times 10^{-5}$ (↓) in new environments $\mathcal{E}_{\text{ad}}$ (*Adaptation*) on *Lotka-Volterra*. Best for each setting in **bold**.

	Number of adaptation trajectories N_{ad}		
	1	5	10
MAML	3150±940	239±16	173±10
LEADS	47.61±12.47	19.89±7.23	19.42±3.52
CoDA-ℓ_1	**1.24**±0.20	**1.21**±0.18	**1.20**±0.17

Table 4. Parameter estimation MAPE (↓) for CoDA-ℓ_1 on LV ($\#\mathcal{E}_{\text{tr}} = 9$), GS ($\#\mathcal{E}_{\text{tr}} = 4$) and NS ($\#\mathcal{E}_{\text{tr}} = 5$).

	In-convex-hull		Out-of-convex-hull		Overall
	MAPE (%)	$\#\mathcal{E}_{\text{ad}}$	MAPE (%)	$\#\mathcal{E}_{\text{ad}}$	MAPE (%)
LV	0.15±0.11	625	0.73±1.33	1976	0.59±1.33
GS	0.37±0.25	625	0.74±0.67	1976	0.65±0.62
NS	0.10±0.08	40	0.51±0.35	41	0.30±0.33

lecting the useful adaptation subspace via $\ell_{1,2}$ regularization, offering a more flexible approach to induce sparsity.

In a second study in Section 5.5, we analyze the impact on MAPE of the dimension of context vectors d_ξ for CoDA-ℓ_1. We recall that d_ξ upper-bounds the dimension of the adaptation subspace $\mathcal{W}$ and was cross-validated in Table 1. In the following, d_p is the number of parameters that vary across environments. We illustrate the effect of the cross-validation on MAPE for $d_p = 2$ on LV and GO as in Section 5.5 and additionally for $d_p = 4$ on LV. We observe in Section 5.5 that the minimum of MAPE is reached for $d_\xi = d_p$ with two regimes: when $d_\xi < d_p$, performance decreases as some system dimensions cannot be learned; when $d_\xi > d_p$, performance degrades slightly as unnecessary directions of variations are added, increasing the hypothesis search space. This study shows the validity of the low-rank assumption and illustrates how the unknown d_p can be recovered through cross-validation.

5.7. Sample Efficiency

We handled originally one-shot adaptation ($N_{\text{ad}} = 1$), the most challenging setting. We vary the number of adaptation trajectories N_{ad} on *Lotka-Volterra* in Table 3. With more trajectories, performance improves significantly for MAML; moderately for LEADS; while it remains flat for CoDA. This highlights CoDA's sample-efficiency and meta-overfitting for GBML (Mishra et al., 2018).

5.8. Parameter Estimation

We use CoDA to perform parameter estimation, leveraging the links between learned context and system parameters.

5.8.1. Empirical Observations

In Figure 5a (left), we visualize on LV the learned context vectors ξ^e (red) and the system parameters p^e (black), $\forall e \in \mathcal{E}_{\text{tr}} \cup \mathcal{E}_{\text{ad}}$. We observe empirically a linear bijection between these two sets of vectors. Such a correspondence being learned on the training environments, we can use the correspondence to verify if it still applies to new adaptation environments. Said otherwise, we can check if our model is able to infer the true parameters for new environments.

We evaluate in Table 4 the parameter estimation MAPE over LV, GS and NS. Figure 5 displays estimated parameters along estimation MAPE. Experimentally, we observe low MAPE inside and even outside the convex-hull of training environments. Thus, CoDA identifies accurately the unknown system parameters with little supervision.

5.8.2. Theoretical Motivation

We justify these empirical observations theoretically in Proposition 3 under the following conditions:

Assumption 1. *The dynamics in $\mathcal{F}$ are linear w.r.t. inputs and system parameters.*

Assumption 2. *Dynamics model g, hypernet A are linear.*

Assumption 3. *$\forall e \in \mathcal{E}$, parameters $p^e \in \mathbb{R}^{d_p}$ are unique.*

Assumption 4. *Context vectors have dimension $d_\xi = d_p$.*

Assumption 5. *The system parameters p of all dynamics f in a basis $\mathcal{B}$ of $\mathcal{F}$ are known.*

Proposition 3 (Identification under linearity. Proof in Appendix C)**.** *Under Assumptions 1 to 5, system parameters are perfectly identified on new environments if the dynamics model g and hypernetwork A satisfy $\forall f \in \mathcal{B}$ with system parameter p, $g_{A(p)} = f$.*

Intuitively, Proposition 3 says that given some observations representative of the degrees of variation of the data (a basis of $\mathcal{F}$) and given the system parameters for these observations (Assumption 5), we are guaranteed to recover the parameters of new environments for a family systems. This strong guarantee requires strong conditions. Assumptions 1 and 2 state that the systems should be linear w.r.t. inputs and that the dynamics model should be linear too. Linearity of the hypernetwork is not an issue as detailed in Section 3.3. Assumption 3 applies to several real-world systems used in our experiments (cf. Appendix C lemmas 1 and 2). Assumption 4 is not restrictive as we showed that d_p is recovered through cross-validation (Section 5.5).

We propose an extension of Proposition 3 in Proposition 4 to nonlinear systems w.r.t. inputs and nonlinear dynamics model g. This alleviates the linearity assumption in Assumptions 1 and 2 and better fits our experimental setting.

Proposition 4 (Local identification under non-linearity. Proof in Appendix C)**.** *For linearly parametrized systems, nonlinear w.r.t. inputs and nonlinear dynamics model g_θ with parameters output by a linear hypernetwork A, $\exists \alpha > 0$ s.t. system parameters are perfectly identified $\forall e \in \mathcal{E}$ where $\|\xi^e\| \leq \alpha$ if $\forall f \in \mathcal{B}$ with parameter p, $g_{A(\alpha \frac{p}{\|p\|})} = f$.*

Proposition 4 states that system parameters are recovered for environments with context vectors of small norm, under a rescaling condition on true system parameters. Proposition 4 explains why estimation error increases when system parameters differ greatly from training ones, as these systems are more likely to violate the norm condition.

6. Related Work

We review Out-of-Distribution (OoD), Multi-Task Learning (MTL) and meta-learning methods and their existing extensions to dynamical systems.

Learning in Multiple Environments OoD methods extend the ERM objective to learn domain invariants e.g. via robust optimization (Sagawa et al., 2020) or Invariant Risk Minimization (IRM) (Arjovsky et al., 2019; Krueger et al., 2021). However, they are not adapted to our problem as an unique model is learned. CoDA is closer to meta-learning and MTL. A standard meta-learning approach is gradient-based meta-learning (GBML), which learns a model initialisation through bi-level optimization. GBML can then adapt to a new task with few gradient steps. The standard GBML method is MAML (Finn et al., 2017), extended in various work. ANIL (Raghu et al., 2020) restricts meta-learning to the last layer of a classifier while other work improve adaptation by preconditioning the gradient (Lee & Choi, 2018; Flennerhag et al., 2020; Park et al., 2019) e.g. Meta-SGD (Li et al., 2017) learns dimension-wise inner-loop learning rates. Contextual meta-learning approaches in Zintgraf et al. (2019); Garnelo et al. (2018) partition parameters into context parameters, adapted on each task, and meta-trained parameters, shared across tasks. CoDA follows the same objective of learning a low-dimensional representation of each task but generalizes these approaches with hypernetworks. For MTL, a standard approach is hard-parameter sharing which shares earlier layers of the network (Caruana, 1997). Several extensions were proposed to learn more efficiently from a set of related tasks (Rebuffi et al., 2017; 2018). Yet, MTL does not address adaptation to new tasks, which is the focus of CoDA. Some extensions have also considered this problem, mainly for classification (Wang et al., 2021a; Requeima et al., 2019).

Generalization for Dynamical Systems Only few work have considered generalization for dynamical systems. LEADS (Yin et al., 2021a) is a MTL approach that performs adaptation in functional space. CoDA operates in parameter space, making adaptation more expressive and efficient, and scales better with the number of environments as it does not require training a full new network per environment as LEADS does. A second work is DyAd (Wang et al., 2021c), a context-aware meta-learning method. DyAd adapts the dynamics model by decoding a time-invariant context, obtained by encoding observed states. However, unlike CoDA, DyAd uses weak supervision obtained from physics quantities to supervise the encoder, which may not always be possible. Moreover, it performs AdaIN modulation (instance normalization + FiLM), a particular case of hypernetwork decoding, which performed worse than CoDA in our experiments.

7. Conclusion

We introduced CoDA, a new framework to learn context-informed data-driven dynamics models on multiple environments. CoDA generalizes with little retraining and few data to new related physical systems and outperforms prior methods on several real-world nonlinear dynamics. Many promising applications of CoDA are possible, notably for spatiotemporal problems, e.g. partially observed systems, reinforcement learning, or NN-based simulation.

Acknowledgement We acknowledge the financial support from DL4CLIM ANR-19-CHIA-0018-01, DEEPNUM ANR-21-CE23-0017-02, OATMIL ANR-17-CE23-0012, RAIMO ANR-20-CHIA-0021-01 and LEAUDS ANR-18-CE23-0020.

References

Antoniou, A., Edwards, H., and Storkey, A. J. How to train your MAML. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=HJGven05Y7. (p. 5)

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *CoRR*, abs/1907.02893, 2019. URL http://arxiv.org/abs/1907.02893. (p. 9)

Bertinetto, L., Henriques, J. F., Torr, P., and Vedaldi, A. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyxnZh0ct7. (p. 14)

Caruana, R. Multitask learning. *Machine Learning*, 28(1): 41–75, 1997. (pp. 9 and 14)

Chen, R. T. Q. torchdiffeq, 2021. URL https://github.com/rtqichen/torchdiffeq. (p. 5)

Chen, Y., Friesen, A. L., Behbahani, F., Doucet, A., Budden, D., Hoffman, M., and de Freitas, N. Modular meta-learning with shrinkage. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 2858–2869. Curran Associates, Inc., 2020a. (p. 7)

Chen, Z., Zhang, J., Arjovsky, M., and Bottou, L. Symplectic recurrent neural networks. In *International Conference on Learning Representations*, 2020b. URL https://openreview.net/forum?id=BkgYPREtPr. (p. 1)

Clavera, I., Nagabandi, A., Liu, S., Fearing, R. S., Abbeel, P., Levine, S., and Finn, C. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyztsoC5Y7. (p. 2)

Courtier, P., Thépaut, J.-N., and Hollingsworth, A. A strategy for operational implementation of 4D-Var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, 120(519):1367–1387, 1994. (p. 1)

Daniels, B. C. and Nemenman, I. Efficient inference of parsimonious phenomenological models of cellular dynamics using s-systems and alternating regression. *PLOS ONE*, 10(3):1–14, 03 2015. (pp. 6, 17, and 18)

de Bézenac, E., Pajot, A., and Gallinari, P. Deep learning for physical processes: Incorporating prior scientific knowledge. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=By4HsfWAZ. (p. 1)

Duraisamy, K., Iaccarino, G., and Xiao, H. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51:357–377, 2019. (p. 1)

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 06–11 Aug 2017. URL http://proceedings.mlr.press/v70/finn17a.html. (pp. 2, 6, 9, and 14)

Flennerhag, S., Rusu, A. A., Pascanu, R., Visin, F., Yin, H., and Hadsell, R. Meta-learning with warped gradient descent. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkeiQlBFPB. (p. 9)

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7. (p. 5)

Fresca, S., Manzoni, A., Dedè, L., and Quarteroni, A. Deep learning-based reduced order models in cardiac electrophysiology. *PloS one*, 15(10):e0239416–e0239416, 10 2020. doi: 10.1371/journal.pone.0239416. URL https://pubmed.ncbi.nlm.nih.gov/33002014. (p. 1)

Garnelo, M., Rosenbaum, D., Maddison, C., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y. W., Rezende, D. J., and Eslami, S. M. A. Conditional neural processes. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1690–1699. PMLR, 2018. URL http://proceedings.mlr.press/v80/garnelo18a.html. (pp. 5 and 9)

Goyal, A., Lamb, A., Zhang, Y., Zhang, S., Courville, A., and Bengio, Y. Professor forcing: A new algorithm for training recurrent networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 4608–4616, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819. (p. 5)

Greydanus, S., Dzamba, M., and Yosinski, J. Hamiltonian neural networks. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 15353–15363, 2019. URL https://proceedings.neurips.cc/paper/2019/file/26cd8ecadce0d4efd6cc8a8725cbd1f8-Paper.pdf. (p. 1)

Gur-Ari, G., Roberts, D. A., and Dyer, E. Gradient descent happens in a tiny subspace, 2019. URL https://openreview.net/forum?id=ByeTHsAqtX. (p. 5)

Ha, D., Dai, A. M., and Le, Q. V. Hypernetworks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=rkpACe1lx. (p. 4)

Hairer, E., Nørsett, S. P., and Wanner, G. *Solving Ordinary Differential Equations I: Nonstiff problems*. Springer, Berlin, second edition, 2000. (p. 5)

Kalman, R. E. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82 (1):35–45, 03 1960. ISSN 0021-9223. doi: 10.1115/1.3662552. URL https://doi.org/10.1115/1.3662552. (p. 1)

Kim, H., Mnih, A., Schwarz, J., Garnelo, M., Eslami, A., Rosenbaum, D., Vinyals, O., and Teh, Y. W. Attentive neural processes. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkE6PjC9KX. (p. 5)

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980. (pp. 6 and 18)

Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., and Hoyer, S. Machine learning accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118, 2021. URL https://www.pnas.org/content/118/21/e2101784118. (p. 1)

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Priol, R. L., and Courville, A. Out-of-distribution generalization via risk extrapolation (REx), 2021. URL https://arxiv.org/pdf/2003.00688.pdf. (p. 9)

Lee, K., Maji, S., Ravichandran, A., and Soatto, S. Meta-learning with differentiable convex optimization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 10657–10665. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.01091. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Lee_Meta-Learning_With_Differentiable_Convex_Optimization_CVPR_2019_paper.html. (p. 14)

Lee, K., Seo, Y., Lee, S., Lee, H., and Shin, J. Context-aware dynamics model for generalization in model-based reinforcement learning. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5757–5766. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/lee20g.html. (p. 2)

Lee, Y. and Choi, S. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning*, pp. 2933–2942, 2018. (p. 9)

Li, C., Farkhoor, H., Liu, R., and Yosinski, J. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018a. URL https://openreview.net/forum?id=ryup8-WCW. (p. 5)

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018b. (pp. 4 and 5)

Li, Z., Zhou, F., Chen, F., and Li, H. Meta-SGD: Learning to learn quickly for few shot learning. *CoRR*, abs/1707.09835, 2017. URL http://arxiv.org/abs/1707.09835. (pp. 7 and 9)

Li, Z., Kovachki, N. B., Azizzadenesheli, K., liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO. (pp. 1, 6, and 18)

Long, Z., Lu, Y., Ma, X., and Dong, B. PDE-Net: Learning PDEs from data. In Dy, J. G. and Krause, A. (eds.), *Pro-*

ceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80 of *Proceedings of Machine Learning Research*, pp. 3214–3222. PMLR, 2018. URL http://proceedings.mlr.press/v80/long18a.html. (p. 5)

Lotka, A. Elements of physical biology. *Nature*, 116 (2917):461–461, 1925. (pp. 6 and 17)

Mishra, N., Rohaninejad, M., Chen, X., and Abbeel, P. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1DmUzWAW. (pp. 2 and 8)

Neic, A., Campos, F. O., Prassl, A. J., Niederer, S. A., Bishop, M. J., Vigmond, E. J., and Plank, G. Efficient computation of electrograms and ecgs in human whole heart simulations using a reaction-eikonal model. *J. Comput. Phys.*, 346:191–211, 2017. doi: 10.1016/j.jcp.2017.06.020. URL https://doi.org/10.1016/j.jcp.2017.06.020. (p. 1)

Park, J. J., Florence, P., Straub, J., Newcombe, R. A., and Lovegrove, S. DeepSDF: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 165–174. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00025. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Park_DeepSDF_Learning_Continuous_Signed_Distance_Functions_for_Shape_Representation_CVPR_2019_paper.html. (p. 9)

Pearson, J. E. Complex patterns in a simple system. *Science*, 261(5118):189–192, 1993. doi: 10.1126/science.261.5118.189. URL https://www.science.org/doi/abs/10.1126/science.261.5118.189. (pp. 6 and 18)

Perez, E., Strub, F., de Vries, H., Dumoulin, V., and Courville, A. C. FiLM: Visual reasoning with a general conditioning layer. In McIlraith, S. A. and Weinberger, K. Q. (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 3942–3951. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16528. (pp. 7 and 14)

Raghu, A., Raghu, M., Bengio, S., and Vinyals, O. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgMkCEtPB. (pp. 7, 9, and 14)

Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125. (p. 5)

Ramachandran, P., Zoph, B., and Le, Q. V. Searching for activation functions, 2018. URL https://openreview.net/forum?id=SkBYYyZRZ. (p. 18)

Rebuffi, S., Bilen, H., and Vedaldi, A. Efficient parametrization of multi-domain deep neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 8119–8127. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00847. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Rebuffi_Efficient_Parametrization_of_CVPR_2018_paper.html. (p. 9)

Rebuffi, S.-A., Bilen, H., and Vedaldi, A. Learning multiple visual domains with residual adapters. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. (p. 9)

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204, 2019. (p. 1)

Requeima, J., Gordon, J., Bronskill, J., Nowozin, S., and Turner, R. E. Fast and flexible multi-task classification using conditional neural adaptive processes. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 7957–7968. 2019. URL https://proceedings.

neurips.cc/paper/2019/file/1138d90ef0a0848a542e57d1595f58ea-Paper.pdf. (p. 9)

Ruder, S. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098, 2017. URL http://arxiv.org/abs/1706.05098. (p. 14)

Rusu, A. A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., and Hadsell, R. Meta-learning with latent embedding optimization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=BJgklhAcK7. (p. 7)

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS. (p. 9)

Shaier, S., Raissi, M., and Seshaiyer, P. Data-driven approaches for predicting spread of infectious diseases through DINNs: Disease informed neural networks. 2021. (p. 1)

Sirignano, J. and Spiliopoulos, K. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018. ISSN 10902716. doi: 10.1016/j.jcp.2018.08.029. (p. 1)

Stokes, G. G. On the Effect of the Internal Friction of Fluids on the Motion of Pendulums. *Transactions of the Cambridge Philosophical Society*, 9:8, January 1851. (pp. 6 and 18)

Thrun, S. and Pratt, L. Y. Learning to learn: Introduction and overview. In Thrun, S. and Pratt, L. Y. (eds.), *Learning to Learn*, pp. 3–17. Springer, 1998. ISBN 978-1-4613-7527-2. doi: 10.1007/978-1-4615-5529-2_1. URL https://doi.org/10.1007/978-1-4615-5529-2_1. (p. 2)

Vogels, T., Karimireddy, S. P., and Jaggi, M. Powersgd: Practical low-rank gradient compression for distributed optimization. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/d9fbed9da256e344c1fa46bb46c34c5f-Paper.pdf. (p. 5)

Wandel, N., Weinmann, M., and Klein, R. Learning incompressible fluid dynamics from scratch - towards fast, differentiable fluid models that generalize. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=KUDUoRsEphu. (p. 1)

Wang, H., Zhao, H., and Li, B. Bridging multi-task learning and meta-learning: Towards efficient training and effective adaptation. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10991–11002. PMLR, 2021a. URL http://proceedings.mlr.press/v139/wang21ad.html. (p. 9)

Wang, J., Lan, C., Liu, C., Ouyang, Y., and Qin, T. Generalizing to unseen domains: A survey on domain generalization. In Zhou, Z. (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pp. 4627–4635. ijcai.org, 2021b. doi: 10.24963/ijcai.2021/628. URL https://doi.org/10.24963/ijcai.2021/628. (p. 1)

Wang, R., Walters, R., and Yu, R. Meta-learning dynamics forecasting using task inference. *CoRR*, abs/2102.10271, 2021c. URL https://arxiv.org/abs/2102.10271. (pp. 2 and 9)

Yin, Y., Ayed, I., de Bézenac, E., Baskiotis, N., and Gallinari, P. LEADS: Learning dynamical systems that generalize across environments. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021a. URL https://openreview.net/forum?id=HD6CxZtbmIx. (pp. 2, 7, 9, 14, and 18)

Yin, Y., Le Guen, V., Dona, J., de Bézenac, E., Ayed, I., Thome, N., and Gallinari, P. Augmenting physical models with deep networks for complex dynamics forecasting. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124012, dec 2021b. doi: 10.1088/1742-5468/ac3ae5. URL https://doi.org/10.1088/1742-5468/ac3ae5. (pp. 1 and 5)

Zintgraf, L., Shiarli, K., Kurin, V., Hofmann, K., and Whiteson, S. Fast context adaptation via meta-learning. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7693–7702. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/zintgraf19a.html. (pp. 5, 7, 9, and 14)

Generalizing to New Physical Systems via Context-Informed Dynamics Model
Supplementary Material

A. Discussion

We discuss in more details the originality and differences of CoDA w.r.t. several Multi-Task Learning (MTL) and gradient-based or contextual meta-learning methods illustrated in Figure 6. We consider CAVIA (Zintgraf et al., 2019), MAML (Finn et al., 2017), ANIL (Raghu et al., 2020), hard-parameter sharing MTL (Caruana, 1997; Ruder, 2017), LEADS (Yin et al., 2021a).

A.1. Adaptation Rule

We compare the adaptation rule in Eq. (4) w.r.t. these work.

GBML Given k gradient steps, MAML defines

$$\theta^e = \theta^c + (-\eta \sum_{i=0}^{k} \nabla_\theta \mathcal{L}(\theta_i^e, \mathcal{D}^e)) \tag{12}$$

$$\text{where } \begin{cases} \theta_{i+1}^e = \theta_i^e - \eta \nabla_\theta \mathcal{L}(\theta_i^e, \mathcal{D}^e) & i > 0 \\ \theta_0^e = \theta^c & i = 0 \end{cases}$$

With $\delta\theta^e \triangleq -\eta \sum_{i=0}^{k} \nabla_\theta \mathcal{L}(\theta_i^e, \mathcal{D}^e)$, Eq. (4) thus includes MAML. ANIL and related GBML methods (Lee et al., 2019; Bertinetto et al., 2019) restrict Eq. (12) to parameters of the final layer, while remaining parameters are shared.

MTL MTL models can be identified to Eq. (12). They fix $\theta^c \triangleq \mathbf{0}$, removing the ability of performing fast adaptation as parameters are retrained from scratch instead of being initialized to θ^c. Hard-parameter sharing MTL restricts the sum in Eq. (12) to the final layer, as ANIL. LEADS sums the outputs of a shared and an environment specific network, thus splits parameters into two independent blocks that do not share connections.

A.2. Decoding for Context-Informed Adaptation

We show that conditioning strategies in contextual meta-learning for decoding context vectors ξ^e into $\delta\theta^e$ are a special case of hypernetwork-decoding. The two main approaches are conditioning via concatenation and conditioning via feature modulation a.k.a. FiLM (Perez et al., 2018).

A.2.1. Conditioning via Concatenation

We show that conditioning via concatenation is equivalent to a linear hypernetwork $A_\phi : \xi^e \mapsto W\xi^e + \theta^c$ with $\phi = \{\theta^c, W\}$ that only predicts the bias of the first layer of g_θ.

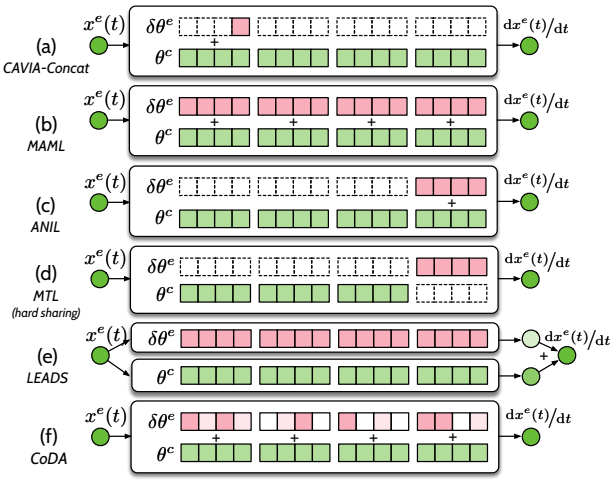


Figure 6. Illustration of representative baselines for multi-environment learning. Shared parameters are blue, environment-specific parameters are red. (a) CAVIA-Concat acts upon the bias of the first layer with conditioning via concatenation. (b) MAML acts upon all parameters without penalization nor prior structure information. (c) ANIL restricts meta-learning to the final layer. (d) Hard-sharing MTL train the final layer from scratch, while the remaining is a *hard-shared*. (e) LEADS sums the output of a common and a environment-specific network. (f) CoDA acts upon a subspace of the parameter space with a locality constraint.

We assume that g_θ has N layers and analyze the output of the first layer of g_θ, omitting the nonlinearity, when the input $x \in \mathbb{R}^{d_x}$ in an environment $e \in \mathcal{E}$ is concatenated to a context vector $\xi^e \in \mathbb{R}^{d_\xi}$. We denote $x \| \xi^e$ the concatenated vector, n_h the number of hidden units of the first layer, $W^1 \in \mathbb{R}^{n_h \times (d_x + d_\xi)}$ and $b^1 \in \mathbb{R}^{n_h}$ the weight matrix and bias term of the first layer, $W^2, \cdots, W^N$ and $b^2, \cdots, b^N$ those of the following layers. The output of the first layer is

$$y^1 = W^1 \cdot x \| \xi^e + b^1$$

We split W^1 along rows into two weight matrices, $W_x^1 \in \mathbb{R}^{n_h \times d_x}$ and $W_\xi^1 \in \mathbb{R}^{n_h \times d_\xi}$ s.t.

$$y^1 = W_x^1 \cdot x + W_\xi^1 \cdot \xi^e + b^1$$

$b_\xi^1 \triangleq W_\xi^1 \cdot \xi^e + b^1$ does not depend on x and corresponds to an environment-specific bias. Thus, concatenation is in-

cluded in Eq. (4) when

$$\theta^c \triangleq \{W_x^1, b^1, W^2, b^2, \cdots, W^N, b^N\}$$
$$\delta\theta^e \triangleq \{\ 0\ , b_\xi^1,\ 0\ , 0\ , \cdots,\ 0\ , 0\}$$

where $\delta\theta^e$ is decoded via a hypernetwork with parameters $\{\theta^c, W \triangleq (0, W_\xi^1, 0, \cdots, 0)\}$.

A.2.2. Conditioning via Feature Modulation

We show that conditioning via FiLM is equivalent to a linear hypernetwork $A_\phi : \xi^e \mapsto W\xi^e + \theta^c$ with $\phi = \{\theta^c, W\}$ that only predicts the batch norm (BN) statistics of g_θ.

For simplicity, we focus on a single BN layer and denote $\{h_i\}_{i=1}^M$, M feature maps output by preceding convolutional layers. These feature maps are first normalized then rescaled with an affine transformation. Rescaling is similar to a FiLM layer that transforms linearly $\{h_i\}_{i=1}^M$ with:

$$\forall i \in \{1, \cdots, M\}, \text{FiLM}(h_i) = \gamma_i \odot h_i + \beta$$

where $\gamma, \beta \in \mathbb{R}^M$ are output by a NN f_ψ conditioned on the context vectors ξ^e i.e. $[\gamma, \beta] = f_\psi(\xi^e)$. In general, f_ψ is linear s.t. $f_\psi(\xi^e) \triangleq W_\xi \xi^e + b_\xi$, with $\psi = \{W_\xi, b_\xi\}$. Then $\gamma = W_\xi^\gamma \xi^e + b_\xi^\gamma, \beta = W_\xi^\beta \xi^e + b_\xi^\beta$.

Thus, for this layer, modulation is included in Eq. (4) when

$$\delta\theta^e \triangleq W\xi^e = \{W_\xi^\gamma \xi^e, W_\xi^\beta \xi^e\}$$
$$\theta^c \triangleq b_\xi = \{b_\xi^\gamma, b_\xi^\beta\}$$

where $\delta\theta^e$ is decoded via hypernetwork $f_\psi \triangleq A_\phi$ with parameters $\phi = \{\theta^c \triangleq b_\xi, W \triangleq W_\xi\}$.

B. Proofs

Proposition 2. *Given a class of linearly parametrized dynamics $\mathcal{F}$ with d_p varying parameters, $\forall \theta^c \in \mathbb{R}^{d_\theta}$, subspace $\mathcal{G}_{\theta^c}$ in Definition 1 is low-dimensional and satisfies $\dim(\mathcal{G}_{\theta^c}) \leq d_p \ll d_\theta$.*

Proof. We define the linear mapping $\psi : p \in \mathbb{R}^{d_p} \to f \in \mathcal{F}$ from parameters to dynamics s.t. $\psi(\mathbb{R}^{d_p}) = \mathcal{F}$. Given this linear mapping, we first prove the following lemma: $\dim(\mathcal{F}) \leq d_p$. The proof is based on surjectivity of ψ onto $\mathcal{F}$, given by definition. We define $\{b_i\}_{i=1}^{d_p}$ a basis of $\mathbb{R}^{d_p}$. Given $f \in \mathcal{F}$, $\exists p \in \mathbb{R}^{d_p}, \psi(p) = f$. We note $p = \sum_{i=1}^{d_p} \lambda_i b_i$ where $\forall i, \lambda_i \in \mathbb{R}$. Then $\psi(p) = \sum_{i=1}^{d_p} \lambda_i \psi(b_i)$. We extract a basis from $\{\psi(b_i)\}_{i=1}^{d_p}$ and denote $d_f \leq d_p$ the number of elements in this basis. This basis forms a basis of $\mathcal{F}$ i.e. $d_f = \dim(\mathcal{F}) \leq d_p$.

Now, given $\theta \in \mathbb{R}^{d_\theta}$ and $f^e \in \mathcal{F}$. We precise that given a (probability) measure $\rho_\mathcal{X}$ on $\mathcal{X} \subset \mathbb{R}^d$, the function space $\mathcal{F} \subset L^2(\rho_x, \mathbb{R}^d)$, then

$$\mathcal{L}(\theta, \mathcal{D}^e) \triangleq \int_\mathcal{X} \|(f^e - g_\theta)(x)\|_2^2 \mathrm{d}\rho_\mathcal{X}(x) = \|f^e - g_\theta\|_2^2$$

The gradient of $\mathcal{L}(\theta, \mathcal{D}^e)$ is then

$$\begin{aligned}\nabla_\theta \mathcal{L}(\theta^c, \mathcal{D}^e) &= \nabla_\theta \int_\mathcal{X} \|f^e(x) - g_{\theta^c}(x)\|_2^2 \mathrm{d}\rho_\mathcal{X}(x) \\ &= \int_\mathcal{X} \nabla_\theta \|f^e(x) - g_{\theta^c}(x)\|_2^2 \mathrm{d}\rho_\mathcal{X}(x) \\ &= -2 \int_\mathcal{X} \mathbf{J}_\theta g_{\theta^c}(x)^\top (f^e(x) - g_{\theta^c}(x)) \mathrm{d}\rho_\mathcal{X}(x) \\ &= -2 D_\theta g_{\theta^c}^\top (f^e - g_{\theta^c})\end{aligned}$$

where $\mathbf{J}_\theta g_\theta(x)$ is the Jacobian matrix of g_{θ^c} w.r.t. θ at point x. $\theta \mapsto D_\theta g_{\theta^c}$ is the differential of g_θ. Note that $D_\theta g_{\theta^c} : \mathbb{R}^{d_\theta} \to \mathcal{F}$ is a linear map (analogue of Jacobian matrix). $D_\theta g_{\theta^c}^\top : \mathcal{F}^\star \to \mathbb{R}^{d_\theta}$ denotes its adjoint (analogue of transposed matrix), which is also a linear map.

As $\mathcal{G}_{\theta^c} \subseteq Im(D_\theta g_\theta^\top)$, then according to Rank-nullity theorem, $\dim(\mathcal{G}_{\theta^c}) \leq \dim(Im(D_\theta g_{\theta^c}^\top)) = \dim(\mathcal{F}) - \dim(Ker(D_\theta g_{\theta^c}^\top)) \leq \dim(\mathcal{F}) \leq d_p$. □

Proposition 1. *Given $\{\theta^c, W\}$ fixed, if $\|\cdot\| = \ell_2$, then Eq. (8) is quadratic. If $\lambda' W^\top W$ or $\bar{H}^e(\theta^c) = W^\top \nabla_\theta^2 \mathcal{L}(\theta^c, \mathcal{D}^e) W$ are invertible then $\bar{H}^e(\theta^c) + \lambda' W^\top W$ is invertible except for a finite number of λ' values. The problem in Eq. (8) is then also convex and admits an unique solution, $\{\xi^{e\star}\}_{e \in \mathcal{E}_{ad}}$. With $\lambda' \triangleq 2\lambda$,*

$$\xi^{e*} = -\Big(\bar{H}^e(\theta^c) + \lambda' W^\top W\Big)^{-1} W^\top \nabla_\theta \mathcal{L}(\theta^c, \mathcal{D}^e)$$

$\bar{H}^e(\theta^c) + \lambda' W^\top W$ is invertible $\forall \lambda'$ except a finite number of values if $\bar{H}^e(\theta^c)$ or $\lambda' W^\top W$ is invertible.

Proof. When $\|\cdot\| = \ell_2$, we consider the following second order Taylor expansion of $\mathcal{L}_\mathrm{r}(\theta, \mathcal{D}^e) \triangleq \mathcal{L}(\theta, \mathcal{D}^e) + \lambda \|\theta - \theta^c\|_2^2$ at θ^c, where $\delta\theta^e = \theta - \theta^c = W\xi^e$.

$$\begin{aligned}\mathcal{L}_\mathrm{r}(\theta^c + \delta\theta^e, \mathcal{D}^e) = \mathcal{L}(\theta^c, \mathcal{D}^e) + \nabla_\theta \mathcal{L}(\theta^c, \mathcal{D}^e)^\top \delta\theta^e + \\ \frac{1}{2} \delta\theta^{e\top} \Big(\nabla_\theta^2 \mathcal{L}(\theta^c, \mathcal{D}^e) + 2\lambda \mathrm{Id}\Big) \delta\theta^e + o(\|\delta\theta^e\|_2^3) \quad (13)\end{aligned}$$

With $\delta\theta^e = W\xi^e$, we expand Eq. (13) into

$$\begin{aligned}\mathcal{L}_\mathrm{r}(\theta^c + W\xi^e, \mathcal{D}^e) = \mathcal{L}(\theta^c, \mathcal{D}^e) + (W^\top \nabla_\theta \mathcal{L}(\theta^c, \mathcal{D}^e))^\top \xi^e \\ + \frac{1}{2} \xi^{e\top} (W^\top \nabla_\theta^2 \mathcal{L}(\theta^c, \mathcal{D}^e) W + 2\lambda W^\top W) \xi^e + o(\|\delta\theta^e\|_2^3)\end{aligned}$$

i.e. with $\bar{H}^e(\theta^c) = W^\top \nabla_\theta^2 \mathcal{L}(\theta^c, \mathcal{D}^e) W$ and $\lambda' = 2\lambda$

$$\begin{aligned}\mathcal{L}_\mathrm{r}(\theta^c + W\xi^e, \mathcal{D}^e) = \mathcal{L}(\theta^c, \mathcal{D}^e) + (W^\top \nabla_\theta \mathcal{L}(\theta^c, \mathcal{D}^e))^\top \xi^e \\ + \frac{1}{2} \xi^{e\top} \Big(\bar{H}^e(\theta^c) + \lambda' W^\top W\Big) \xi^e + o(\|\delta\theta^e\|_2^3) \quad (14)\end{aligned}$$

Eq. (14) is quadratic. If $\bar{H}^e(\theta^c) + \lambda' W^\top W$ is invertible, then the problem is also convex with unique solution

$$\xi^{e*} = -\Big(\bar{H}^e(\theta^c) + \lambda' W^\top W\Big)^{-1} W^\top \nabla_\theta \mathcal{L}(\theta^c, \mathcal{D}^e)$$

$\bar{H}^e(\theta^c)$ and $\lambda' W^\top W$ are two square matrices. The application $p : \lambda' \mapsto \det(\bar{H}^e(\theta^c) + \lambda' W^\top W)$ is well-defined and forms a continuous polynomial. Thus either it equals zero or it has a finite number of roots. If $\bar{H}^e(\theta^c)$ or $\lambda' W^\top W$ is invertible, then $p(0) = \det(\bar{H}^e(\theta^c)) \neq 0$ or $p(\infty) \sim \det(\lambda' W^\top W) \neq 0$. Thus $p \neq 0$ has a finite number of roots i.e. $\bar{H}^e(\theta^c) + \lambda' W^\top W$ is invertible $\forall \lambda'$ except a finite number of values corresponding to the roots of p. □

C. System Parameter Estimation

Proposition 3. *Under Assumptions 1 to 5, system parameters are perfectly identified on new environments if the dynamics model g and hypernetwork A satisfy $\forall f \in \mathcal{B}$ with system parameter p, $g_{A(p)} = f$.*

Proof. We define the linear mapping $\psi : p \in \mathbb{R}^{d_p} \to f \in \mathcal{F}$ from parameters to dynamics s.t. $\psi(\mathbb{R}^{d_p}) = \mathcal{F}$ (Assumption 1). Unicity of parameters (Assumption 3) implies that ψ is bijective with inverse ψ^{-1}, thus $\dim(\mathcal{F}) = \dim(\mathbb{R}^{d_p}) = d_p$. Given a basis $\mathcal{B} = \{f_i\}_{i=1}^{d_p}$ of $\mathcal{F}$, we denote $p_i = \psi^{-1}(f_i)$. We fix g, A s.t. $\forall i \in \{1, ..., d_p\}, g_{A(p_i)} = f_i = \psi(p_i)$. This is possible as f_i and g are linear w.r.t. inputs (Assumptions 1 and 2) and p_i are known (Assumption 5).

g, A are linear (Assumption 2), thus $g_{A(\cdot)}$ is linear with inputs in $\mathbb{R}^{d_\xi}$. Then, $\dim(\mathrm{Im}(g_{A(\cdot)})) \leq d_\xi$. Moreover, $\forall i \in \{1, ..., d_p\}, f_i \in \mathrm{Im}(g_{A(\cdot)})$, thus $\mathcal{F} \subset \mathrm{Im}(g_{A(\cdot)})$ i.e. $d_p \leq \dim(\mathrm{Im}(g_{A(\cdot)}))$. Thus, $d_p \leq \dim(\mathrm{Im}(g_{A(\cdot)})) \leq d_\xi$. Assumption 4 states that $d_\xi = d_p$, s.t. $\dim(\mathrm{Im}(g_{A(\cdot)})) = d_p$. As $\mathcal{F} \subset \mathrm{Im}(g_{A(\cdot)})$ and $\dim(\mathcal{F}) = \dim(\mathrm{Im}(g_{A(\cdot)}))$, $\mathcal{F} = \mathrm{Im}(g_{A(\cdot)})$ i.e. $g_{A(\cdot)}$ is surjective onto $\mathcal{F}$. As $\dim(\mathcal{F}) = d_\xi$, the dimension of the input space, $g_{A(\cdot)}$ is bijective.

By bijectivity of ψ, $\{p_i\}_{i=1}^{d_p}$ forms a basis of $\mathbb{R}^{d_p}$. $g_{A(\cdot)}$ and ψ map this basis to the same basis $\{f_i\}_{i=1}^{d_p}$ of $\mathcal{F}$. As both mappings are bijective, this implies that $g_{A(\cdot)} = \psi(\cdot)$. This means that $\forall e \in \mathcal{E}, {g_A}^{-1}(f^e) = \psi^{-1}(f^e)$ i.e. system parameters p^e are recovered. □

Proposition 4. *For linearly parametrized systems, nonlinear w.r.t. inputs and nonlinear dynamics model g_θ with parameters output by a linear hypernetwork A, $\exists \alpha > 0$ s.t. system parameters are perfectly identified $\forall e \in \mathcal{E}$ where $\|\xi^e\| \leq \alpha$ if $\forall f \in \mathcal{B}$ with parameter p, $g_{A(\alpha \frac{p}{\|p\|})} = f$.*

Proof. On environment $e \in \mathcal{E}$, g_{θ^e} is differentiable w.r.t. $\theta^e = A(\xi^e) = \theta^c + W\xi^e \in \mathbb{R}^{d_\theta}$. We perform a first order Taylor expansion of $g_{A(\cdot)}$ around $\mathbf{0}$. We note $\alpha > 0$, s.t. $\forall \xi^e \in \mathbb{R}^{d_\xi}$ that satisfy $\|\xi^e\| < \alpha$, we have $g_{\theta^e} = g_{\theta^c} + \nabla_\theta g_{\theta^c} W \xi^e$. $g_{A(\cdot)}$ is then linear in the neighborhood of $\mathbf{0}$ defined by α. $\forall i \in [\![1, d_p]\!]$, $\alpha \frac{p_i}{\|p_i\|}$ belongs to this neighborhood s.t. the proof of Proposition 3 applies to this neighborhood if $\forall i \in [\![1, d_p]\!], g_{A(\alpha \frac{p_i}{\|p_i\|})} = f_i$, where $\mathcal{B} = \{f_i\}_{i=1}^{d_p}$ is a basis of $\mathcal{F}$. □

We now show the validity of the unicity condition (Assumption 3) for two linearly parametrized systems.

Lemma 1. *There is an unique set of parameters in $\mathbb{R}^4$ for a Lotka-Volterra (`LV`) system.*

Proof. With $\psi : c \triangleq (\alpha, \beta, \delta, \gamma) \mapsto [\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} \alpha x - \beta xy \\ \delta xy - \gamma y \end{pmatrix}]$ a surjective linear mapping from $\mathbb{R}^4$ to $\mathcal{F}$ (all `LV` systems are parametrized). Injectivity of ψ i.e. $\psi(c_1) = \psi(c_2) \iff c_1 = c_2$ will imply bijectivity i.e. unicity of parameters for a `LV` system. As ψ is linear, injectivity is equivalent to $\psi(c) = 0 \iff c = 0$, shown below:

$$\begin{aligned}
\psi(c) = 0 &\iff \forall \begin{pmatrix} x \\ y \end{pmatrix}, \begin{pmatrix} x(\alpha - \beta y) \\ (\delta x - \gamma) y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\
&\iff \forall \begin{pmatrix} x \\ y \end{pmatrix}, \begin{pmatrix} \alpha - \beta y \\ \delta x - \gamma \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\
&\iff c = (\alpha, \beta, \delta, \gamma) = (0, 0, 0, 0)
\end{aligned}$$

□

Lemma 2. *There is an unique set of parameters in $\mathbb{R}^{d+1}$, where d is the grid size, for a Navier-Stokes (`NS`) system.*

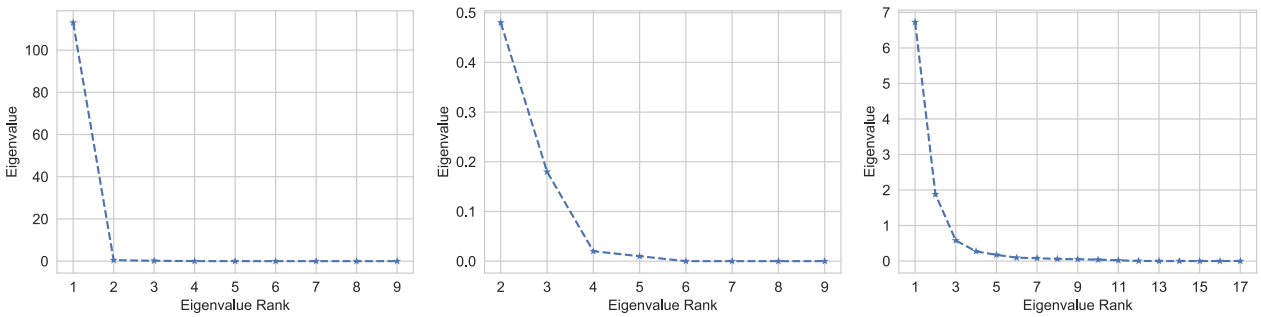
Proof. With $\psi : c \triangleq (\nu, f) \mapsto \Big[w \mapsto -v\nabla w + \nu \Delta w + f\Big]$, a surjective linear mapping from $\mathbb{R}^{d+1}$ to $\mathcal{F}$ (all `NS` systems are parametrized), bijectivity of ψ is induced by injectivity i.e. $\psi(c_1) = \psi(c_2) \iff c_1 = c_2$, shown below:

$$\begin{aligned}
&\psi(c_1) = \psi(c_2) \\
\iff &\forall w, -v\nabla w + \nu_1 \Delta w + f_1 = -v\nabla w + \nu_2 \Delta w + f_2 \\
\iff &\forall w, (\nu_1 - \nu_2)\Delta w = -(f_1 - f_2) \\
\iff &(\nu_1, f_1) = (\nu_2, f_2) \iff c_1 = c_2
\end{aligned}$$

□

D. Low-Rank Assumption

When the systems are nonlinearly parametrized, we show empirically with Figure 7 that the low-rank assumption is still reasonable for two different systems.



(a) GO: k_1 and K_1 vary across $\mathcal{E}$. (b) Sin.

Figure 7. Ranked singular values of the gradients across environments $\mathcal{E}_{\text{tr}}$, $\mathcal{G}_{\theta^c}$ for CoDA-ℓ_1.

Glycolitic-Oscillator (GO) We consider the Glycolitic-Oscillator system (GO), described in Appendix F.1, which is nonlinear w.r.t. K_1. We vary parameters k_1, K_1 in Eq. (16) across environments. We observe in Figure 7a that there are three main gradient directions with SVD. The first is the most significant one while the second and third ones are orders of magnitude smaller.

Sinusoidal (Sin) We consider a sinusoidal family of functions $S(n) = \{f : \mathbb{R} \to \mathbb{R} | f(x) = \sum_{i=1}^{N} \lambda_i \sin(\omega_i x + \phi_i)\}$ (Sin). We sample 20 environments that correspond each to different amplitudes (uniformly sampled in $[0, 1]$), frequencies (uniformly sampled in $[0, 10]$)) and phases (uniformly sampled in $[0, 3.14]$). We depict in Figure 7b the evaluation of the singular values at initialization. Figure 7b shows that the number of directions to consider for convergence is small and that a single direction accounts for a significant amount of the variance in the gradients. This corroborates the low-rank assumption.

E. Locality Constraint

We derive the upper-bounds to $\|\cdot\|$ for two variations.

$\|\cdot\| = \ell_2$: we apply triangle inequality to obtain $\Omega = \ell_2^2$

$$\|W\xi^e\|_2^2 \leq \|W\|_2^2 \|\xi^e\|_2^2$$

$\|\cdot\| = \ell_1$: we apply Cauchy-Schwartz inequality to obtain $\Omega(W) = \ell_{1,2}(W) \triangleq \sum_{i=1}^{d_\theta} \|W_{i,:}\|_2$

$$\|W\xi^e\|_1 = \sum_{i=1}^{d_\theta} |W_{i,:}\xi^e| \leq \|\xi^e\|_2 \sum_{i=1}^{d_\theta} \|W_{i,:}\|_2$$

Eq. (11) minimizes the log of the above upper-bounds.

F. Experimental Settings

We present in Appendix F.1 the equations and the data generation specificities for all considered dynamical systems.

F.1. Dynamical Systems

Lotka-Volterra (LV, Lotka, 1925) The system describes the interaction between a prey-predator pair in an ecosystem, formalized into the following ODE:

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= \alpha x - \beta xy \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= \delta xy - \gamma y \end{aligned} \tag{15}$$

where x, y are respectively the quantity of the prey and the predator, $\alpha, \beta, \delta, \gamma$ define how two species interact.

We generate trajectories on a temporal grid with $\Delta t = 0.5$ and temporal horizon $T = 10$. We sample on each training environment $N_{\text{tr}} = 4$ initial conditions for training from a uniform distribution $p(X_0) = \text{Unif}([1, 3]^2)$. We sample for evaluation 32 initial conditions from $p(X_0)$. Across environments, $\alpha = 0.5, \gamma = 0.5$. For training, we consider $\#\mathcal{E}_{\text{tr}} = 9$ environments with parameters $\beta, \delta \in \{0.5, 0.75, 1.0\}^2$. For adaptation, we consider $\#\mathcal{E}_{\text{ad}} = 4$ environments with parameters $\beta, \delta \in \{0.625, 1.125\}^2$.

Glycolytic-Oscillator (GO, Daniels & Nemenman, 2015) GO describes yeast glycolysis dynamics with the ODE:

$$\begin{aligned} \frac{\mathrm{d}S_1}{\mathrm{d}t} &= J_0 - \frac{k_1 S_1 S_6}{1 + (1/K_1^q) S_6^q} \\ \frac{\mathrm{d}S_2}{\mathrm{d}t} &= 2\frac{k_1 S_1 S_6}{1 + (1/K_1^q) S_6^q} - k_2 S_2 (N - S_5) - k_6 S_2 S_5 \\ \frac{\mathrm{d}S_3}{\mathrm{d}t} &= k_2 S_2 (N - S_5) - k_3 S_3 (A - S_6) \\ \frac{\mathrm{d}S_4}{\mathrm{d}t} &= k_3 S_3 (A - S_6) - k_4 S_4 S_5 - \kappa (S_4 - S_7) \\ \frac{\mathrm{d}S_5}{\mathrm{d}t} &= k_2 S_2 (N - S_5) - k_4 S_4 S_5 - k_6 S_2 S_5 \\ \frac{\mathrm{d}S_6}{\mathrm{d}t} &= -2\frac{k_1 S_1 S_6}{1 + (1/K_1^q) S_6^q} + 2k_3 S_3 (A - S_6) - k_5 S_6 \\ \frac{\mathrm{d}S_7}{\mathrm{d}t} &= \psi \kappa (S_4 - S_7) - k S_7 \end{aligned} \tag{16}$$

where $S_1, S_2, S_3, S_4, S_5, S_6, S_7$ represent the concentrations of 7 biochemical species. We generate trajectories on a temporal grid with $\Delta t = 0.05$ and temporal horizon $T = 1$. We sample on each training environment $N_{\text{tr}} = 32$ initial conditions for training from a uniform distribution $p(X_0)$ defined in Table 2 in (Daniels & Nemenman, 2015). Across environments, $J_0 = 2.5, k_2 = 6, k_3 = 16, k_4 = 100, k_5 = 1.28, k_6 = 12, q = 4, N = 1, A = 4, \kappa = 13, \psi = 0.1, k = 1.8$. For training, we consider $\#\mathcal{E}_{\text{tr}} = 9$ environments with parameters $k_1 \in \{100, 90, 80\}, K_1 \in \{1, 0.75, 0.5\}$. For adaptation, we consider $\#\mathcal{E}_{\text{ad}} = 4$ environments with parameters $k_1 \in \{85, 95\}, K_1 \in \{0.625, 0.875\}$.

Gray-Scott (GS, Pearson, 1993) The PDE descibes a reaction-diffusion system with complex spatiotemporal patterns through the following 2D PDE:

$$\begin{aligned} \frac{\partial u}{\partial t} &= D_u \Delta u - uv^2 + F(1-u) \\ \frac{\partial v}{\partial t} &= D_v \Delta v + uv^2 - (F+k)v \end{aligned} \tag{17}$$

where u, v represent the concentrations of two chemical components in the spatial domain S with periodic boundary conditions. D_u, D_v denote the diffusion coefficients respectively for u, v and F, k are the reaction parameters.

We generate trajectories on a temporal grid with $\Delta t = 40$ and temporal horizon $T = 400$. S is a 2D space of dimension 32×32 with spatial resolution of $\Delta s = 2$. We define initial conditions $(u_0, v_0) \sim p(X_0)$ by uniformly sampling three two-by-two squares in S. These squares trigger the reactions. $(u_0, v_0) = (1-\epsilon, \epsilon)$ with $\epsilon = 0.05$ inside the squares and $(u_0, v_0) = (0, 1)$ outside the squares. We sample on each training environment $N_{\text{tr}} = 1$ initial conditions for training. Across environments, $D_u = 0.2097, D_v = 0.105$. For training, we consider $\#\mathcal{E}_{\text{tr}} = 4$ environments with parameters $F \in \{0.30, 0.39\}, k \in \{0.058, 0.062\}$. For adaptation, we consider $\#\mathcal{E}_{\text{ad}} = 4$ environments with parameters $F \in \{0.33, 0.36\}, k \in \{0.59, 0.61\}$.

Navier-Stokes (NS, Stokes, 1851) NS describes the dynamics of incompressible flows with the 2D PDE:

$$\begin{aligned} &\frac{\partial w}{\partial t} = -v \nabla w + \nu \Delta w + f \text{ where } w = \nabla \times v \\ &\nabla v = 0 \end{aligned} \tag{18}$$

where v is the velocity field, $w = \nabla \times v$ is the vorticity. Both v, w lie in a spatial domain S with periodic boundary conditions, ν is the viscosity and f is the constant forcing term in the domain S. We generate trajectories on a temporal grid with $\Delta t = 1$ and temporal horizon $T = 10$. S is a 2D space of dimension 32×32 with spatial resolution of $\Delta s = 1$. We sample on each training environment $N_{\text{tr}} = 16$ initial conditions for training from $p(X_0)$ as in Li et al. (2021). Across environments, $f(X, Y) = 0.1(\sin(2\pi(X+Y)) + \cos(2\pi(X+Y)))$. For training, we consider $\#\mathcal{E}_{\text{tr}} = 5$ environments with parameters $\nu \in \{8 \cdot 10^{-4}, 9 \cdot 10^{-4}, 1.0 \cdot 10^{-3}, 1.1 \cdot 10^{-3}, 1.2 \cdot 10^{-3}\}$. For adaptation, we consider $\#\mathcal{E}_{\text{ad}} = 4$ environments with parameters $\nu \in \{8.5 \cdot 10^{-4}, 9.5 \cdot 10^{-4}, 1.05 \cdot 10^{-3}, 1.15 \cdot 10^{-3}\}$.

F.2. Implementation and Hyperparameters

Architecture We implement the dynamics model g_θ with the following architectures:

- LV, GO: 4-layer MLPs with hidden layers of width 64.
- GS: 4-layer ConvNet with 64-channel hidden layers, and 3×3 convolution kernels
- NS: Fourier Neural Operator (Li et al., 2021) with 4 spectral convolution layers. 12 frequency modes and hidden layers with width 10.

We apply Swish activation (Ramachandran et al., 2018). The hypernet A is a single affine layer NN.

Optimizer We use the Adam optimizer (Kingma & Ba, 2015) with learning rate 10^{-3} and $(\beta_1, \beta_2) = (0.9, 0.999)$. We apply early stopping. All experiments are performed with a single NVIDIA Titan Xp GPU on an internal cluster. We distribute training by batching together predictions across trajectories to reduce running time. States across batch elements are concatenated.

Hyperparameters We define hyperparameters for the following models: (a) CoDA: • LV: $\lambda_\xi = 10^{-4}$, $\lambda_{\ell_1} = 10^{-6}$, $\lambda_{\ell_2} = 10^{-5}$ • GO: $\lambda_\xi = 10^{-3}$, $\lambda_{\ell_1} = 10^{-7}$, $\lambda_{\ell_2} = 10^{-7}$ • GS: $\lambda_\xi = 10^{-2}$, $\lambda_{\ell_1} = 10^{-5}$, $\lambda_{\ell_2} = 10^{-5}$ • NS: $\lambda_\xi = 10^{-3}$, $\lambda_{\ell_1} = 2 \cdot 10^{-3}$, $\lambda_{\ell_2} = 2 \cdot 10^{-3}$ (b) LEADS: we use the same parameters as Yin et al. (2021a). (c) GBML: the outer-loop learning rate is 10^{-3}, we apply 1-step inner-loop update for training and adaptation to maintain low running times. The inner-loop learning rate for each system is: • LV: 0.1 • GO: 0.01 • GS: 10^{-3} • NS: 10^{-3}. These values are also used to initialize the per–parameter inner-loop learning rate in Meta-SGD.

G. Trajectory Prediction Visualization

We visualize in Figures 8 and 9 the prediction MSE by MAML, LEADS, CAVIA-Concat and CoDA-ℓ_1 along ground truth trajectories on the PDE systems NS and GS. We consider a new test trajectory on an *Adaptation* environment $e \in \mathcal{E}_{\text{ad}}$ with parameters defined in the caption.

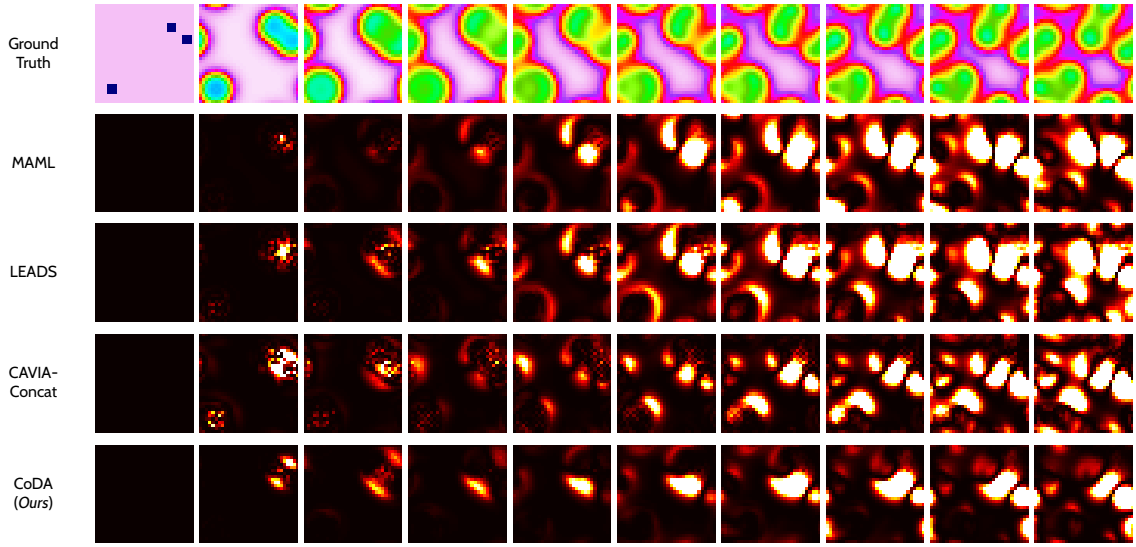


Figure 8. Adaptation to new GS system - $(F, k, D_u, D_v) = (0.033, 0.061, 0.2097, 0.105)$. Ground-truth trajectory and prediction MSE per frame for MAML, LEADS, CAVIA-Concat and CoDA.

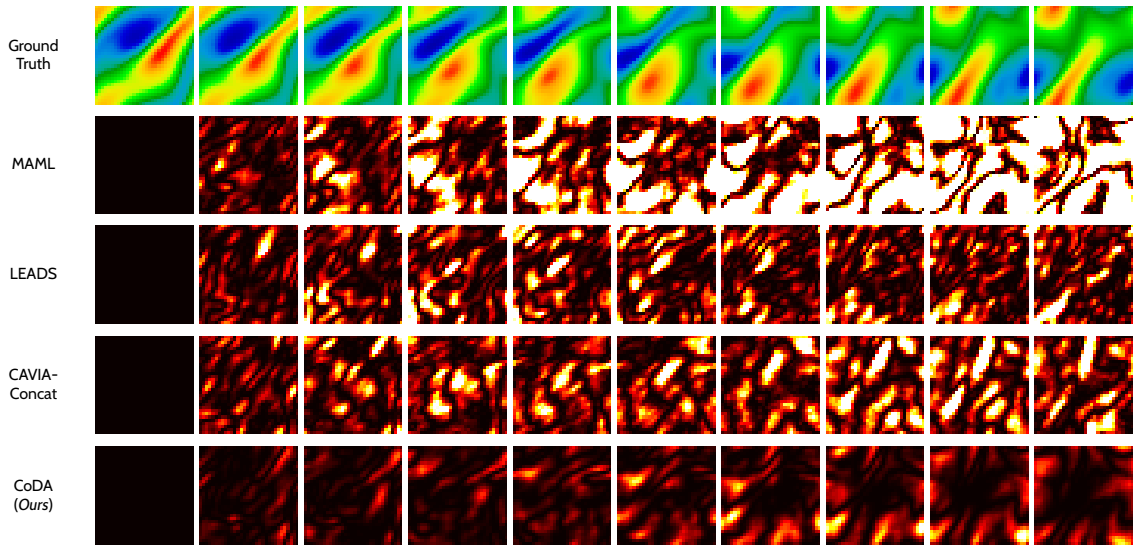


Figure 9. Adaptation to new NS system - $\nu = 1.15 \cdot 10^{-3}$. Ground-truth trajectory and prediction MSE per frame for MAML, LEADS, CAVIA-Concat and CoDA.