



HAL
open science

Generalizing to New Physical Systems via Context-Informed Dynamics Model

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

► **To cite this version:**

Matthieu Kirchmeyer, Yuan Yin, Jérémie Donà, Nicolas Baskiotis, Alain Rakotomamonjy, et al..
Generalizing to New Physical Systems via Context-Informed Dynamics Model. 2022. hal-03547546v1

HAL Id: hal-03547546

<https://hal.science/hal-03547546v1>

Preprint submitted on 31 Jan 2022 (v1), last revised 16 Jun 2022 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Generalizing to New Physical Systems via Context-Informed Dynamics Model

Matthieu Kirchmeyer^{1,2,*} Yuan Yin^{1,*} Jérémie Donà¹ Nicolas Baskiotis¹ Alain Rakotomamonjy^{3,2}
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 to foster fast adaptation and better generalization across environments. It extends the expressivity of existing methods. 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 (Re-

ichstein 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 (Duraismy 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).

On the other hand, 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. Domain generalization methods attempt to cap-

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

ture 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 process 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 like classification or regression on static data or simple sequences. 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 **Context-Informed Dynamics Adaptation (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:

$$\frac{dx(t)}{dt} = f(x(t)), \quad (1)$$

where $t \in \mathbb{R}$ is a time index, $x(t)$ is a time-dependent state in a space \mathcal{X} and $f : \mathcal{X} \rightarrow T\mathcal{X}$ 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} . $\mathcal{X} \subseteq \mathbb{R}^d$ ($d \in \mathbb{N}^*$) 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 $x : I \rightarrow \mathcal{X}$, 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:

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

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 $\mathcal{D}^f(\mathcal{X})$ with the form in Eq. (2). The standard ERM objective considers that all trajectories are i.i.d. Here, we propose a multi-environment learning formulation which considers that observed trajectories of f form an environment $e \in \mathcal{E}$. We denote f^e and $\mathcal{D}^e \triangleq \{\mathcal{D}_i^e\}_{i=1}^N$ the corresponding function and set of trajectories. We assume that we observe a set of known functions in training environments $\mathcal{E}_{\text{tr}}, \{f^e\}_{e \in \mathcal{E}_{\text{tr}}}$. The goal is to

learn g_θ that adapts easily and efficiently to new environments \mathcal{E}_{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 corresponding Mean Squared Error (MSE) loss, over \mathcal{D}^e as

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

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 $\|\cdot\|$:

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

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

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 $\{\theta^{e*}\}_{e \in \mathcal{E}}$ 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 a 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

Solving 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 under scarce data. We propose a hypernetwork-based solution to solve efficiently this problem by operating on a low-dimensional space. It yields fixed-cost adaptation and enables efficient sharing of 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 $\xi^e \in \mathbb{R}^{d_\xi}$. $W = (W_1, \dots, W_{d_\xi}) \in \mathbb{R}^{d_\theta \times d_\xi}$ 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 $\mathcal{W} \triangleq \text{Span}(W_1, \dots, W_{d_\xi})$, 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):

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

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

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

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 $\bar{H}^e(\theta^c) = W^\top \nabla_\theta^2 \mathcal{L}(\theta^c, \mathcal{D}^e)W$ 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, $\{\xi^{e*}\}_{e \in \mathcal{E}_{\text{ad}}}$. With $\lambda' \triangleq 2\lambda$,*

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

Interpretation We now interpret CoDA by visualizing its loss landscape around θ^c in Figure 1, following Li et al. (2018b). We consider the Lotka-Volterra system, described in Section 5.1. Loss values 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 subspace \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 $\bar{H}^e(\theta^c) + \lambda'W^\top W$ 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 is forced to be close to local optimas of all training environments. The final observation illustrates that CoDA is able to find a subspace \mathcal{W} on which there are environment-specific parameters with low loss values i.e. that low-rank adaptation performs well. We provide in Appendix H, some further comparison with the loss landscape of ERM, projected onto the Span of the two principal gradient directions. We show that ERM does not find low loss values, as 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.

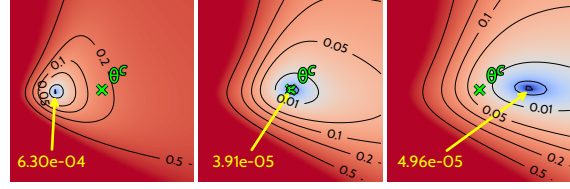


Figure 1. CoDA’s loss landscape centered in θ^c , marked with \times , for 3 environments on the Lotka-Volterra ODE. Loss values are projected onto subspace \mathcal{W} , with $d_\xi = 2$. $\forall e$, \rightarrow points to the local optimum θ^{e*} with loss value reported in yellow.

Definition 1 (Gradient directions). *With \mathcal{L} in Eq. (3), $\forall \theta^c \in \mathbb{R}^{d_\theta}$ parameterizing 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}_{θ^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}_{θ^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 Korteweg–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 g_θ still holds for several systems. This property is comforted by recent studies that highlighted that gradients are low-rank throughout optimization in single-domain settings, meaning 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 w.r.t. related methods, with further details 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) considerably simplifies the complex bi-level optimization problem (Antoniou et al., 2019). Moreover, 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. 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 com-

plexity. 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. 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), defined over N trajectories $\{\mathcal{D}_i^e\}_{i=1}^N$, discretized over a uniform temporal and spatial grid. $\Delta t, \Delta s$ are the temporal and spatial resolutions and T, S the temporal horizon and spatial grid size. Each trajectory includes $\frac{T}{\Delta t} (\frac{S}{\Delta s})^{d_s}$ states, where for PDEs d_s is the spatial dimension and for ODEs $d_s = 0$. With $t_k = k\Delta t$, and s_j the j^{th} spatial coordinate, we denote $x^{e,i}(t_k, s_j)$ the state value in trajectory i from environment e at spatial coordinate s_j and time t_k . Our loss writes as:

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

$$\tilde{x}^{e,i}(t_k) = x^{e,i}(t_{k-1}) + \int_{t_{k-1}}^{t_k} g_\theta(\tilde{x}^{e,i}(\tau)) d\tau \quad (10)$$

where $x(t) = [x(t, s_1), \dots, x(t, s_{(S/\Delta s)^{d_s}})]^\top$ is the state vector over the spatial domain at t . 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:

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

It involves hyperparameters $\lambda_\xi, \lambda_\Omega$ and a norm $\Omega(W)$ which depends on the choice of $\|\cdot\|$ in Eq. (5). We consider two 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. Minimizing $R(W, \xi^e)$ can be interpreted as minimizing an upper-bound to $\|\cdot\|$, derived in appendix E for each variation.

Pseudo-code We solve Eq. (7) for training and Eq. (8) for adaptation using eqs. (10) and (11) and Algorithm 1. We

Algorithm 1 CoDA Pseudo-code

Training:

Input: $\mathcal{E}_{\text{tr}} \subset \mathcal{E}, \{\mathcal{D}^e\}_{e \in \mathcal{E}_{\text{tr}}}$ with $\forall e \in \mathcal{E}_{\text{tr}}, \#\mathcal{D}^e = N$;
 $\pi = \{W, \theta^c, \{\xi^e\}_{e \in \mathcal{E}_{\text{tr}}}\}$ where $W \in \mathbb{R}^{d_\theta \times d_\xi}, \theta^c \in \mathbb{R}^{d_\theta}$
 are randomly initialized and $\forall e \in \mathcal{E}_{\text{tr}}, \xi^e = \mathbf{0} \in \mathbb{R}^{d_\xi}$.

loop

$$\pi \leftarrow \pi - \eta \nabla_\pi \left(\sum_{e \in \mathcal{E}_{\text{tr}}} \sum_{i=1}^N \mathcal{L}(\theta^c + W\xi^e, \mathcal{D}_i^e) + R(W, \xi^e) \right)$$

Adaptation:

Input: $e \in \mathcal{E}_{\text{ad}}; \mathcal{D}^e$ where $\#\mathcal{D}^e = N$;

Trained $W \in \mathbb{R}^{d_\theta \times d_\xi}, \theta^c \in \mathbb{R}^{d_\theta}$ and $\xi^e = \mathbf{0} \in \mathbb{R}^{d_\xi}$.

loop

$$\xi^e \leftarrow \xi^e - \eta \nabla_{\xi^e} \left(\sum_{i=1}^N \mathcal{L}(\theta^c + W\xi^e, \mathcal{D}_i^e) + R(W, \xi^e) \right)$$

back-propagate through the solver’s internals with torchdiffeq (Chen, 2021) and apply exponential Scheduled Sampling (Goyal et al., 2016) to stabilize training.

5. Experiments

We validate our approach on four classes of challenging nonlinear temporal and spatiotemporal physical dynamics, 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 a single trajectory is 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 parameterized systems.

5.2. Baselines

We consider three families of baselines, compared in Appendix Figure 5 and detailed in Section 6. First, Gradient-Based Meta-Learning (GBML) methods MAML (Finn 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-

Table 1. Test MSE (\downarrow) in training environments \mathcal{E}_{tr} (*In-Domain*) and new adaptation environments \mathcal{E}_{ad} (*Adaptation*).

	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 \pm 1.3	3150 \pm 940	57.3 \pm 2.1	1081 \pm 62	3.67 \pm 0.53	2.25 \pm 0.39	68.0 \pm 8.0	51.1 \pm 4.0
ANIL	381 \pm 76	4570 \pm 2390	74.5 \pm 11.5	1688 \pm 226	5.01 \pm 0.80	3.95 \pm 0.11	61.7 \pm 4.3	48.6 \pm 3.2
META-SGD	32.7 \pm 12.6	7220 \pm 4580	42.3 \pm 6.9	1573 \pm 413	2.85 \pm 0.54	2.68 \pm 0.20	53.9 \pm 28.1	44.3 \pm 27.1
LEADS	3.70 \pm 0.27	47.61 \pm 12.47	31.4 \pm 3.3	113.8 \pm 41.5	2.90 \pm 0.76	1.36 \pm 0.43	14.0 \pm 1.55	28.6 \pm 7.23
CAVIA-FiLM	4.38 \pm 1.15	8.41 \pm 3.20	4.44 \pm 1.46	3.87 \pm 1.28	2.81 \pm 1.15	1.43 \pm 1.07	23.2 \pm 12.1	22.6 \pm 9.88
CAVIA-CONCAT	2.43 \pm 0.66	6.26 \pm 0.77	5.09 \pm 0.35	2.37 \pm 0.23	2.67 \pm 0.48	1.62 \pm 0.85	25.5 \pm 6.31	26.0 \pm 8.24
CoDA- ℓ_2	1.52 \pm 0.08	1.82 \pm 0.24	2.45 \pm 0.38	1.98 \pm 0.06	1.01 \pm 0.15	0.77 \pm 0.10	9.40 \pm 1.13	10.3 \pm 1.48
CoDA- ℓ_1	1.35\pm0.22	1.24\pm0.20	2.20\pm0.26	1.86\pm0.29	0.90\pm0.057	0.74\pm0.10	8.35\pm1.71	9.65\pm1.37

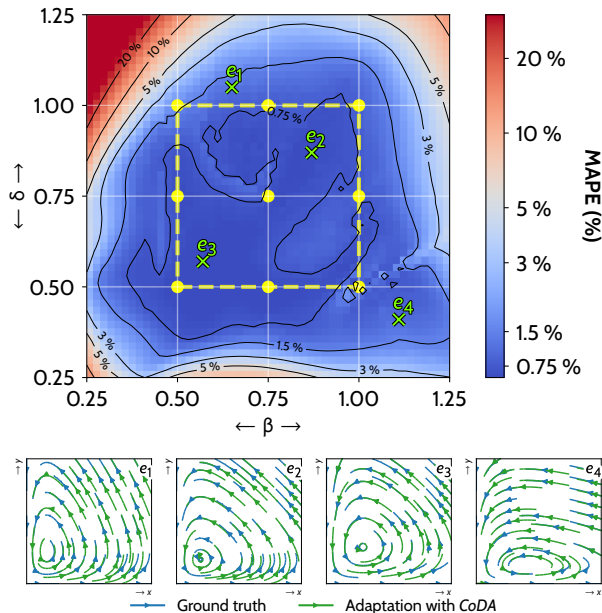


Figure 2. Adaptation results with CoDA- ℓ_1 on LV. Parameters (β, δ) are sampled in $[0.25, 1.25]^2$ on a 51×51 uniform grid, leading to 2601 adaptation environments \mathcal{E}_{ad} . \bullet are training environments \mathcal{E}_{tr} . We report MAPE (\downarrow) across \mathcal{E}_{ad} (Top). On the bottom, we choose four of them (\times , 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).

learning method CAVIA (Zintgraf et al., 2019), with conditioning via concatenation (Concat) or linear modulation of final hidden features (FiLM, Perez et al., 2018). Baselines consider also the loss in Eq. (10).

5.3. Architecture, optimizer and hyperparameters

We use MLPs for ODEs, ConvNets for GS and Fourier Neural Operators (Li et al., 2021) for NS (details in Appendix F.2). Adam optimizer (Kingma & Ba, 2015) is used for all datasets. We tuned d_ξ and observed that $d_\xi = d_p$, the number of system parameters that vary across environments, performed best. This is reported in an ablation study in Section 5.5. Solvers, optimization and regularization hyperparameters are detailed in Appendix F.2.

5.4. Experimental Setting

Each environment $e \in \mathcal{E}$ is defined by system parameters and we denote $p^e \in \mathbb{R}^{d_p}$ those that vary across \mathcal{E} . d_p represents the degrees of variations in \mathcal{F} and is set to $d_p = 2$ for LV, GO, GS and $d_p = 1$ for NS. We define in Appendix F.1 for each system the number of training and adaptation environments ($\#\mathcal{E}_{\text{tr}}$ and $\#\mathcal{E}_{\text{ad}}$) and the corresponding parameters. We also define in Appendix F.1 the number of training trajectories N per environment and the distribution $p(X_0)$ from which are sampled initial conditions.

We perform two types of evaluation: in-domain generalization on \mathcal{E}_{tr} (*In-domain*) and out-of-domain adaptation to new environments \mathcal{E}_{ad} (*Adaptation*). Evaluation is performed on 32 new test trajectories per environment with initial conditions sampled from $p(X_0)$. 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 $\int_{t \in I} \text{MAPE}(\tilde{x}(t), x(t)) dt$, with \tilde{x} in Eq. (10).

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. We visualize trajectories for PDE systems in Appendix G and also notice improvements. Across datasets, all baselines are subject to a drop in performance 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 lo-

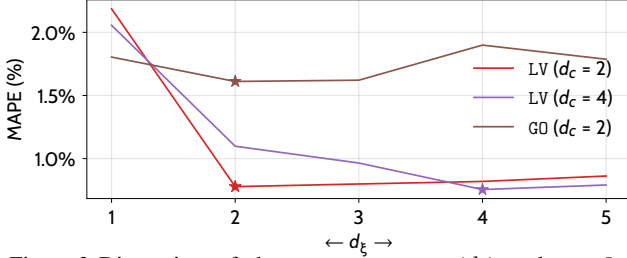


Figure 3. Dimension of the context vectors (d_ξ) and test *In-Domain* MAPE (\downarrow) with CoDA- ℓ_1 . “*” is the minimum of MAPE.

Table 2. ℓ_2 locality constraint and *In-Domain* test MSE (\downarrow).

CoDA	LV ($\times 10^{-5}$)		GO ($\times 10^{-4}$)	
	W/o ℓ_2	WITH ℓ_2	W/o ℓ_2	WITH ℓ_2
FULL	2.28 \pm 0.29	1.52 \pm 0.08	2.98 \pm 0.71	2.45 \pm 0.38
FIRSTLAYER	2.25 \pm 0.29	2.41 \pm 0.23	2.38 \pm 0.71	2.12\pm0.55
LASTLAYER	1.86 \pm 0.24	1.27\pm0.03	28.4 \pm 0.60	28.4 \pm 0.64

cality constraint. CoDA- ℓ_1 is better than CoDA- ℓ_2 as it induces sparsity, further constraining the hypothesis space.

We evaluate in Figure 2 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}_{ad} (Top). We observe three regimes: inside the convex hull of training environments \mathcal{E}_{tr} , MAPE is very low; outside the convex-hull, MAPE remains low in a neighborhood of \mathcal{E}_{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 closely follows the target trajectories.

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 parameterization 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 selecting 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. Parameter estimation

In Figure 4a (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 the 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 3 the parameter estimation MAPE over LV, GS and NS. Figure 4 displays estimated parameters along estimation MAPE for LV and NS. This visualization is provided for GS in Figure 10 (Appendix I). Experimentally, we observe low MAPE inside the convex-hull of training environments and even outside it. This shows that CoDA identifies accurately the unknown system parameters p^e with little supervision.

We justify these empirical observations theoretically in Proposition 3 under the conditions in Assumption 1:

Assumption 1. (a) the dynamics in \mathcal{F} are linear w.r.t. inputs and system parameters, (b) the dynamics model g and hypernetwork A are linear, (c) $\forall e \in \mathcal{E}$, the system parameters of f^e , $p^e \in \mathbb{R}^{d_p}$, are unique, (d) the dimension of context vectors d_ξ is fixed to d_p , (e) the system parameters p_i of dynamic $f_i \in \mathcal{B}$, where \mathcal{B} is a basis of \mathcal{F} , are known.

Proposition 3 (Identification under linearity. Proof in Appendix C). Under Assumption 1, system parameters are perfectly identified on new environments if the dynamics model g and hypernetwork A satisfy $\forall f_i \in \mathcal{B}$, $g_{A(p_i)} = f_i$.

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 (condition (d) in Assumption 1), we are guaranteed to recover the parameters of new environments for a family systems. This strong guarantee requires strong conditions described in Assumption 1. (a), (b) state that the systems should be linear w.r.t. inputs and that the dynam-

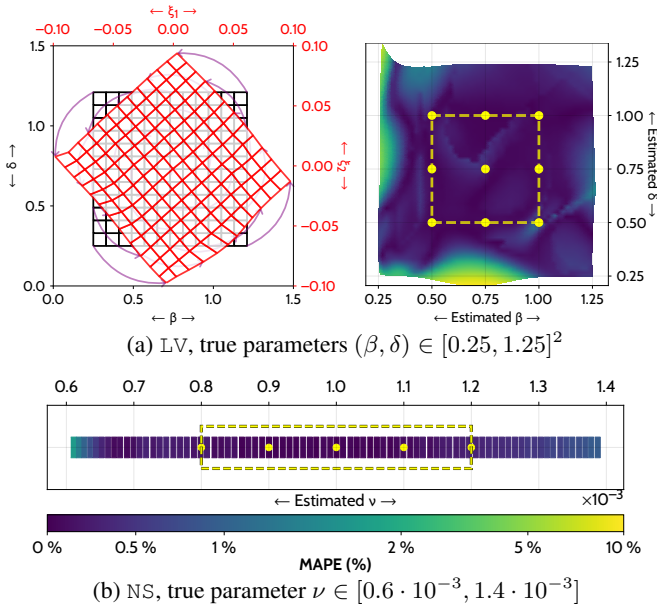


Figure 4. Parameter estimation with CoDA- ℓ_1 in new adaptation environments on LV (a) and NS (b). On LV, we visualize, on the left, context vectors ξ (red) and true parameters (β, δ) (black). On other figures, we visualize estimated parameters with corresponding estimation MAPE (\downarrow). \bullet are training environments \mathcal{E}_{tr} with known parameters. $-\cdot-$ delimits the convex hull of \mathcal{E}_{tr} .

Table 3. Parameter estimation MAPE (\downarrow) 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 (%)
	MAPE (%)	$\#\mathcal{E}_{\text{ad}}$	MAPE (%)	$\#\mathcal{E}_{\text{ad}}$	
LV	0.15 \pm 0.11	625	0.73 \pm 1.33	1976	0.59 \pm 1.33
GS	0.37 \pm 0.25	625	0.74 \pm 0.67	1976	0.65 \pm 0.62
NS	0.10 \pm 0.08	40	0.51 \pm 0.35	41	0.30 \pm 0.33

ics model should be linear too. Linearity of the hypernetwork is not an issue as detailed in Section 3.3. (c) applies to several real-world systems used in our experiments (cf. Appendix C lemmas 1 and 2). (d) is not restrictive as we showed that d_p is recovered through cross-validation (Section 5.5). We propose an extension of Proposition 3 to nonlinear systems w.r.t. inputs and nonlinear dynamics model g in Appendix C Proposition 4. This alleviates the linearity assumption in (a), (b) and fits our experimental setting.

6. Related Work

We review OoD, Multi-Task Learning (MTL) and meta-learning methods and 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.

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>.
- 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>.
- 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>.
- Caruana, R. Multitask learning. *Machine Learning*, 28(1): 41–75, 1997.
- Chen, R. T. Q. `torchdiffeq`, 2021. URL <https://github.com/rtqichen/torchdiffeq>.
- 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.
- 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>.
- 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>.
- 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.
- 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.
- 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>.
- Duraisamy, K., Iaccarino, G., and Xiao, H. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51:357–377, 2019.
- 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 Research*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 06–11 Aug 2017. URL <http://proceedings.mlr.press/v70/finn17a.html>.
- 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>.
- 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>.
- 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>.
- 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>.
- 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.
- 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/hash/26cd8ecadce0d4efd6cc8a8725cbd1f8-Abstract.html>. 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.
- Gur-Ari, G., Roberts, D. A., and Dyer, E. Gradient descent happens in a tiny subspace, 2019. URL <https://openreview.net/forum?id=ByeTHsAqtX>.
- 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=rkpACellx>.
- Hairer, E., Nørsett, S. P., and Wanner, G. *Solving Ordinary Differential Equations I: Nonstiff problems*. Springer, Berlin, second edition, 2000.
- 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>.
- 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>.
- 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>.
- 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>.
- 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>.
- 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. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Lee_Meta-Learning_With_Differentiable_Convex_Optimization_CVPR_2019_paper.html.
- 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>.
- 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.
- 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>.
- 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.
- 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>.
- 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>.
- Long, Z., Lu, Y., Ma, X., and Dong, B. PDE-Net: Learning PDEs from data. 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. 3214–3222.

- PMLR, 2018. URL <http://proceedings.mlr.press/v80/long18a.html>.
- Lotka, A. Elements of physical biology. *Nature*, 116 (2917):461–461, 1925.
- 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>.
- 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 egs 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>.
- 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.
- 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>.
- 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>.
- 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>.
- 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>.
- Ramachandran, P., Zoph, B., and Le, Q. V. Searching for activation functions, 2018. URL <https://openreview.net/forum?id=SkBYyZRZ>.
- 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.
- 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.
- 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.
- 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>.
- Ruder, S. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098, 2017. URL <http://arxiv.org/abs/1706.05098>.
- Rusu, A. A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., and Hadsell, R. Meta-learning with la-

- tent 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>.
- 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>.
- Shaier, S., Raissi, M., and Seshaiyer, P. Data-driven approaches for predicting spread of infectious diseases through DINNs: Disease informed neural networks. 2021.
- 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.
- 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.
- 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.
- 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>.
- 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>.
- 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>.
- 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>.
- 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>.
- 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>.
- Yin, Y., Guen, V. L., 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>.
- 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>.

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 5. 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

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

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

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 $\phi =$

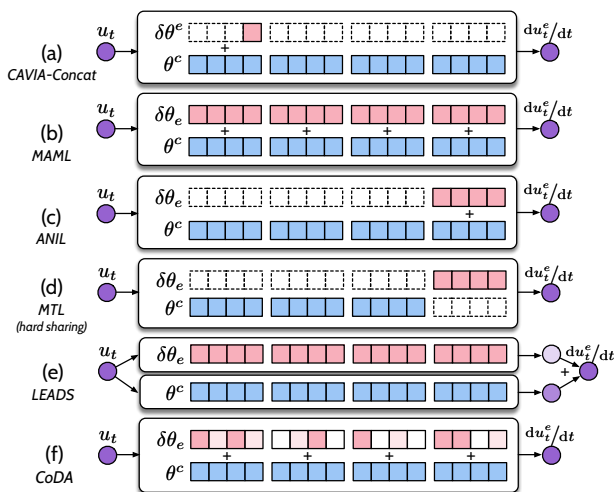


Figure 5. Illustration of representative baselines for multi-environment learning. Shared parameters are blue, environment-specific parameters are red. (a) CAVIA-Concat acts upon the bias of the first layer with conditioning via concatenation. (b) MAML acts upon all parameters without penalization nor 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 an environment-specific network. (f) CoDA acts upon a subspace of the parameter space with a locality constraint.

$\{\theta^c, W\}$ that only predicts the bias of the first layer of g_{θ} .

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 $x \parallel \xi^e$ 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, \dots, W^N and b^2, \dots, b^N those of the following layers. The output of the first layer is

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

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

$$\begin{aligned}\theta^c &\triangleq \{W_x^1, b^1, W^2, b^2, \dots, W^N, b^N\} \\ \delta\theta^e &\triangleq \{0, b_\xi^1, 0, 0, \dots, 0, 0\}\end{aligned}$$

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

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:

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

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

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

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}_{θ^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 $\psi : p \in \mathbb{R}^{d_p} \rightarrow f \in \mathcal{F}$ 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 $f^e \in \mathcal{F}$ and $\theta \in \mathbb{R}^{d_\theta}$,

$$\mathcal{L}(\theta, \mathcal{D}^e) \triangleq \mathbb{E}_{x \in \mathcal{D}^e} \|(f^e - g_\theta)(x)\|_2^2 = \|f^e - g_\theta\|_2^2$$

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

$$\nabla_\theta \mathcal{L}(\theta, \mathcal{D}^e) = -2 \left(\frac{dg_\theta}{d\theta} \right)^\top (f^e - g_\theta)$$

where the adjoint of a linear map h is denoted h^\top . $f \mapsto -2 \left(\frac{dg_\theta}{d\theta} \right)^\top f$ is a linear map as $\theta \mapsto \frac{dg_\theta}{d\theta}$ is linear (differential of g_θ) and the adjoint preserves linearity, s.t. $\dim \text{Span}(\{\nabla_\theta \mathcal{L}(\theta, \mathcal{D}^e)\}_{e \in \mathcal{E}}) \leq \dim(\mathcal{F}) \leq d_p$. \square

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 a unique solution, $\{\xi^{e*}\}_{e \in \mathcal{E}_{ad}}$. With $\lambda' \triangleq 2\lambda$,*

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

$\bar{H}^e(\theta^c) + \lambda' W^\top W$ 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 $\mathcal{L}_{\text{reg}}(\theta, \mathcal{D}^e) = \mathcal{L}(\theta, \mathcal{D}^e) + \lambda \|\theta - \theta^c\|_2^2$ at θ^c , where $\delta\theta^e = \theta - \theta^c = W\xi^e$.

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

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

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

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

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

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

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

$\bar{H}^e(\theta^c)$ 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 . \square

C. System parameter estimation

We show in Proposition 3 that parameters of new systems can be recovered under Assumption 1.

Proposition 3. *Under Assumption 1, system parameters are perfectly identified on new environments if the model g and hypernetwork A satisfy $\forall f_i \in \mathcal{B}, g_{A(p_i)} = f_i$.*

Proof. We define the linear mapping $\psi : p \in \mathbb{R}^{d_p} \rightarrow f \in \mathcal{F}$ from parameters to dynamics s.t. $\psi(\mathbb{R}^{d_p}) = \mathcal{F}$ (Assumption 1 (a)). Unicity of parameters (Assumption 1 (c)) 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. $\forall i \in \{1, \dots, d_p\}, g_{A(p_i)} = f_i = \psi(p_i)$. This is possible as f_i and g are linear w.r.t. inputs (Assumption 1 (a) and (b)) and p_i are known (Assumption 1 (e)).

$\forall i \in \{1, \dots, d_p\}, f_i \in \text{Im}(g_{A(\cdot)})$, thus $\mathcal{F} \subset \text{Im}(g_{A(\cdot)})$ i.e. $d_p \leq \dim(\text{Im}(g_{A(\cdot)}))$. g, A are linear (Assumption 1 (b)), thus $g_{A(\cdot)}$ is linear with inputs in \mathbb{R}^{d_ξ} . Then, $\dim(\text{Im}(g_{A(\cdot)})) \leq d_\xi$ and $d_p \leq \dim(\text{Im}(g_{A(\cdot)})) \leq d_\xi$. $d_\xi = d_p$ (Assumption 1 (d)) i.e. $\dim(\text{Im}(g_{A(\cdot)})) = d_p$. As $\mathcal{F} \subset \text{Im}(g_{A(\cdot)})$, this implies that $\mathcal{F} = \text{Im}(g_{A(\cdot)})$ i.e. $g_{A(\cdot)}$ is surjective onto \mathcal{F} . As $\dim(\mathcal{F}) = d_\xi, g_{A(\cdot)}$ is bijective.

By bijectivity of $\psi, \{p_i\}_{i=1}^{d_p}$ 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. \square

Proposition 4 (Extension to nonlinear dynamics). *For linearly parametrized systems, non-linear w.r.t. inputs and nonlinear dynamics model g_θ where θ is output by a linear hypernetwork $A, \exists \alpha > 0$ s.t. system parameters are perfectly identified on all environments $e \in \mathcal{E}$ that satisfy $\|\xi^e\| \leq \alpha$, if $\forall i \in \llbracket 1, d_p \rrbracket, g_{A(\alpha \frac{p_i}{\|p_i\|})} = f_i$.*

Proof. On environment $e \in \mathcal{E}, g_{\theta^e}$ 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 . $\forall i, \alpha \frac{p_i}{\|p_i\|}$ belongs to this neighborhood s.t. the proof of Proposition 3 applies to this neighborhood if $\forall i \in \llbracket 1, d_p \rrbracket, g_{A(\alpha \frac{p_i}{\|p_i\|})} = f_i$. \square

We now show the validity of the unicity condition (Assumption 1c) 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.

$$\text{With } \psi : c \triangleq (\alpha, \beta, \delta, \gamma) \mapsto \left[\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} \alpha x - \beta xy \\ \delta xy - \gamma y \end{pmatrix} \right]$$

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:

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

\square

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 $\psi : c \triangleq (\nu, f) \mapsto [w \mapsto -v\nabla w + \nu\Delta w + f]$, 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:

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

\square

D. Low-rank assumption

When the systems are nonlinearly parametrized, we show empirically with Figure 6 that the low-rank assumption is still reasonable for two different systems.

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 6 (Left, Middle) 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 $S(n) = \{f : \mathbb{R} \rightarrow \mathbb{R} | f(x) = \sum_{i=1}^N \lambda_i \sin(\omega_i x + \phi_i)\}$ (Sin). We sample 20 environments that correspond

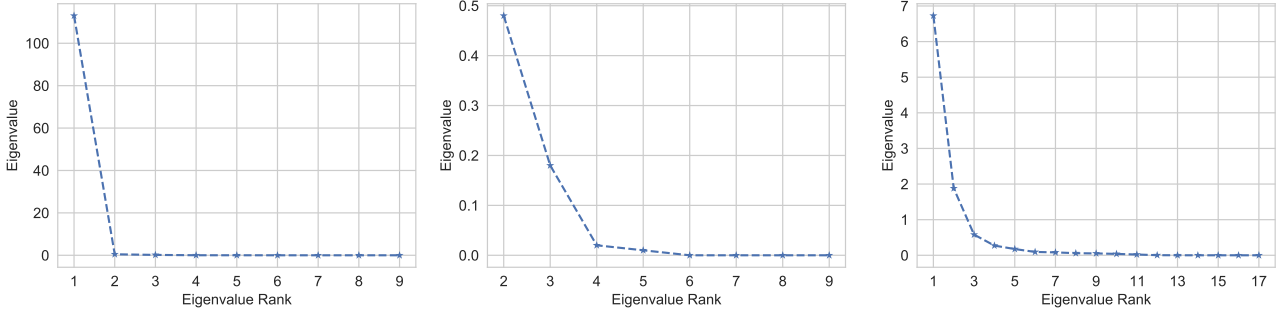


Figure 6. Ranked singular values of the gradients across environments $\mathcal{E}_{tr}, \mathcal{G}_{\theta^e}$ for CoDA- ℓ_1 . On the Left and Middle, we consider GO where k_1 and K_1 vary across \mathcal{E} . On the right, we consider Sin.

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 6 (Right) the evaluation of the singular values at initialization. Figure 6 (Right) 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.

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

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

$$\|\cdot\| = \ell_1: \text{ we apply Cauchy-Schwartz inequality to obtain}$$

$$\Omega(W) = \ell_{1,2}(W) \triangleq \sum_{i=1}^{d_\theta} \|W_{i,:}\|_2$$

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

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:

$$\begin{aligned} \frac{dx}{dt} &= \alpha x - \beta xy \\ \frac{dy}{dt} &= \delta xy - \gamma y \end{aligned} \quad (15)$$

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 environment $N = 4$ 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 $\#\mathcal{E}_{tr} = 9$ environments with parameters $\beta, \delta \in \{0.5, 0.75, 1.0\}^2$. For adaptation, we consider $\#\mathcal{E}_{ad} = 4$ 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:

$$\begin{aligned} \frac{dS_1}{dt} &= J_0 - \frac{k_1 S_1 S_6}{1 + (1/K_1^q) S_6^q} \\ \frac{dS_2}{dt} &= 2 \frac{k_1 S_1 S_6}{1 + (1/K_1^q) S_6^q} - k_2 S_2 (N - S_5) - k_6 S_2 S_5 \\ \frac{dS_3}{dt} &= k_2 S_2 (N - S_5) - k_3 S_3 (A - S_6) \\ \frac{dS_4}{dt} &= k_3 S_3 (A - S_6) - k_4 S_4 S_5 - \kappa (S_4 - S_7) \\ \frac{dS_5}{dt} &= k_2 S_2 (N - S_5) - k_4 S_4 S_5 - k_6 S_2 S_5 \\ \frac{dS_6}{dt} &= -2 \frac{k_1 S_1 S_6}{1 + (1/K_1^q) S_6^q} + 2k_3 S_3 (A - S_6) - k_5 S_6 \\ \frac{dS_7}{dt} &= \psi \kappa (S_4 - S_7) - k S_7 \end{aligned} \quad (16)$$

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 environment $N = 32$ initial conditions for training from a uniform distribution $p(X_0)$ defined in Table 2 in (Daniels & Nemenman, 2015). We sample for adaptation 1 initial condition from $p(X_0)$. 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 $\#\mathcal{E}_{tr} = 9$ 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 describes a reaction-diffusion system with complex spatiotemporal patterns through the following 2D PDE:

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

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 environment $N = 1$ initial conditions for training. We sample for adaptation 1 initial condition. 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:

$$\begin{aligned} \frac{\partial w}{\partial t} &= -v \nabla w + \nu \Delta w + f \\ \nabla v &= 0 \\ w &= \nabla \times v \end{aligned} \quad (18)$$

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 environment $N = 16$ initial conditions for training from $p(X_0)$ as in Li et al. (2021). We sample for adaptation 1 initial condition from $p(X_0)$. 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 and solver We implement the dynamics model g_θ with the following architectures:

- for LV and GO, 4-layer MLPs with 64-dimension hidden layers
- for GS, 4-layer ConvNet with 64-channel hidden layers, and 3×3 convolution kernels
- for GS and a Fourier Neural Operator (Li et al., 2021) with 4 spectral convolution layers for NS. The number of frequency modes is 12 and the hidden layers have 10 dimensions.

We apply Swish activation (Ramachandran et al., 2018) on all architectures and RK4 solver for LV, GS, GO and Euler solver for NS. 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. For GBML methods, we choose a single inner-loop / outer-loop step to maintain low running times. We distribute training by batching together predictions across trajectories to reduce running time. States across batch elements are concatenated.

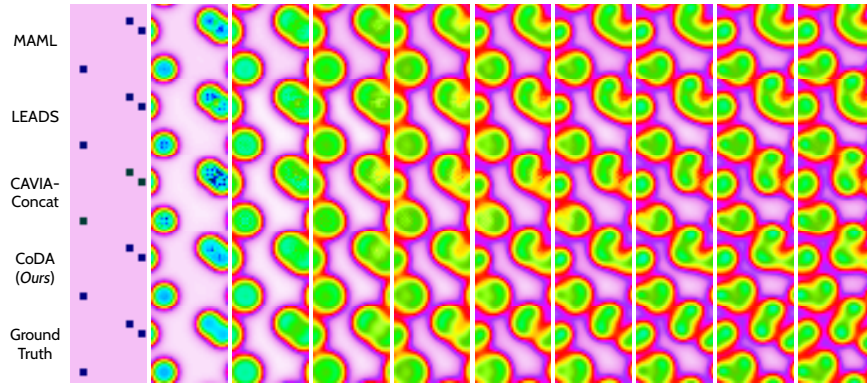
Hyperparameters We define hyperparameters for the following models: (a) CoDA: • LV: $\lambda_\xi = 10^{-4}$, $\lambda_{\ell_1} = 10^{-6}$, $\lambda_{\ell_2} = 10^{-5}$ • GO: $\lambda_\xi = 10^{-3}$, $\lambda_{\ell_1} = 10^{-7}$, $\lambda_{\ell_2} = 10^{-7}$ • GS: $\lambda_\xi = 10^{-2}$, $\lambda_{\ell_1} = 10^{-5}$, $\lambda_{\ell_2} = 10^{-5}$ • 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, and 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

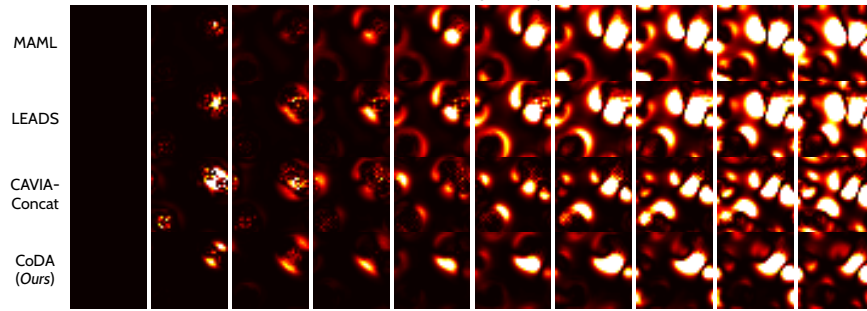
We visualize in Figures 7 and 8 predicted trajectories by MAML, LEADS, CAVIA-Concat and CoDA- ℓ_1 along ground truth trajectories on the PDE systems NS and GS. We consider a new test trajectory on an *Adaptation* environment $e \in \mathcal{E}_{\text{ad}}$ with parameters defined in the caption.

H. Loss landscape visualization

We visualize loss landscapes of CoDA, ERM in Figure 9.

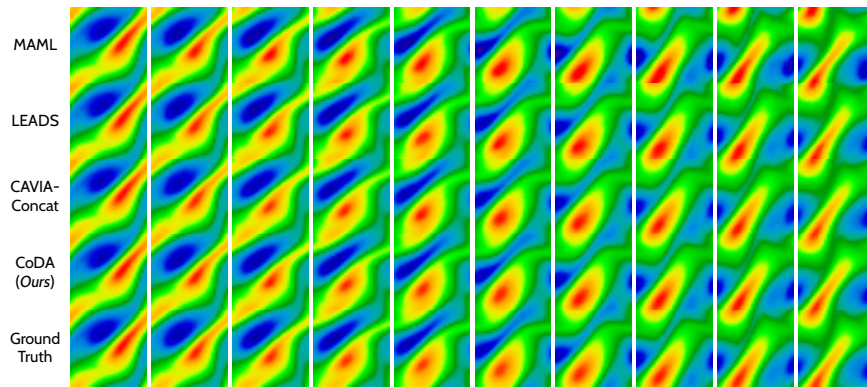


(a) Predicted trajectory

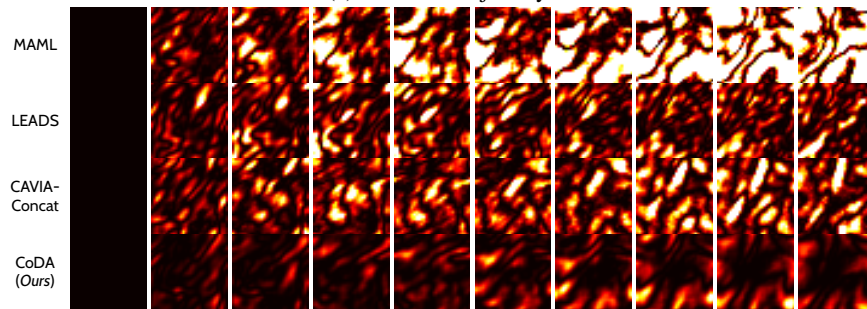


(b) Prediction MSE

Figure 7. Adaptation to new GS system - $(F, k) = (0.033, 0.061)$



(a) Predicted trajectory



(b) Prediction MSE

Figure 8. Adaptation to new NS system - $\nu^e = 1.15 \cdot 10^{-3}$

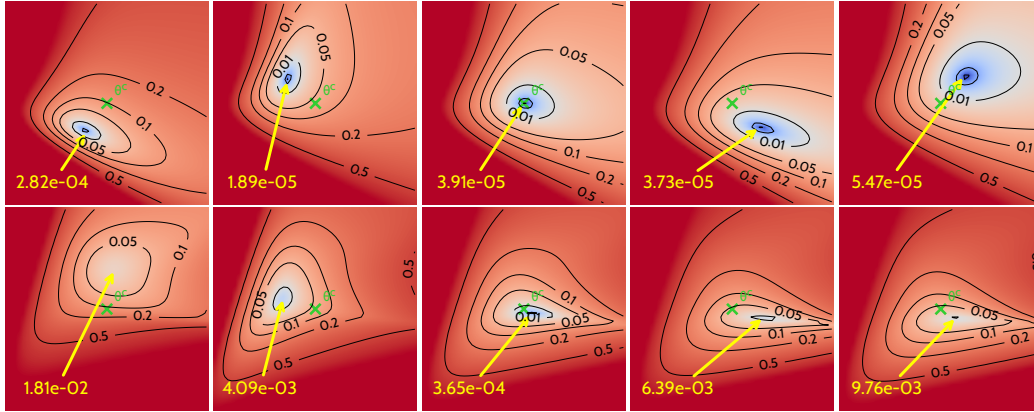


Figure 9. Loss landscapes around θ^c (\times) for 5 training environments on the Lotka-Volterra ODE system for ERM and CoDA. The local optimum θ^{e*} is indicated with \rightarrow . CoDA’s loss (Row 1) is projected onto subspace \mathcal{W} , with $d_\xi = 2$. ERM’s loss (Row 2) is projected onto the two principal directions of the gradients computed with Singular Value Decomposition.

I. System parameter estimation

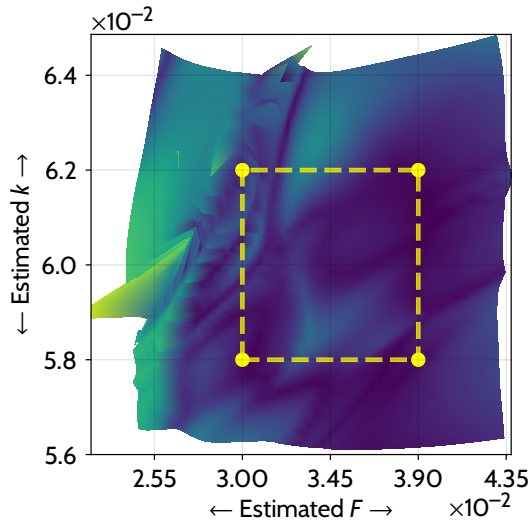


Figure 10. Parameter estimation MAPE (\downarrow) and estimated parameters on GS over environments defined by $(F, k) \in [0.0225, 0.0435] \times [0.056, 0.064]$