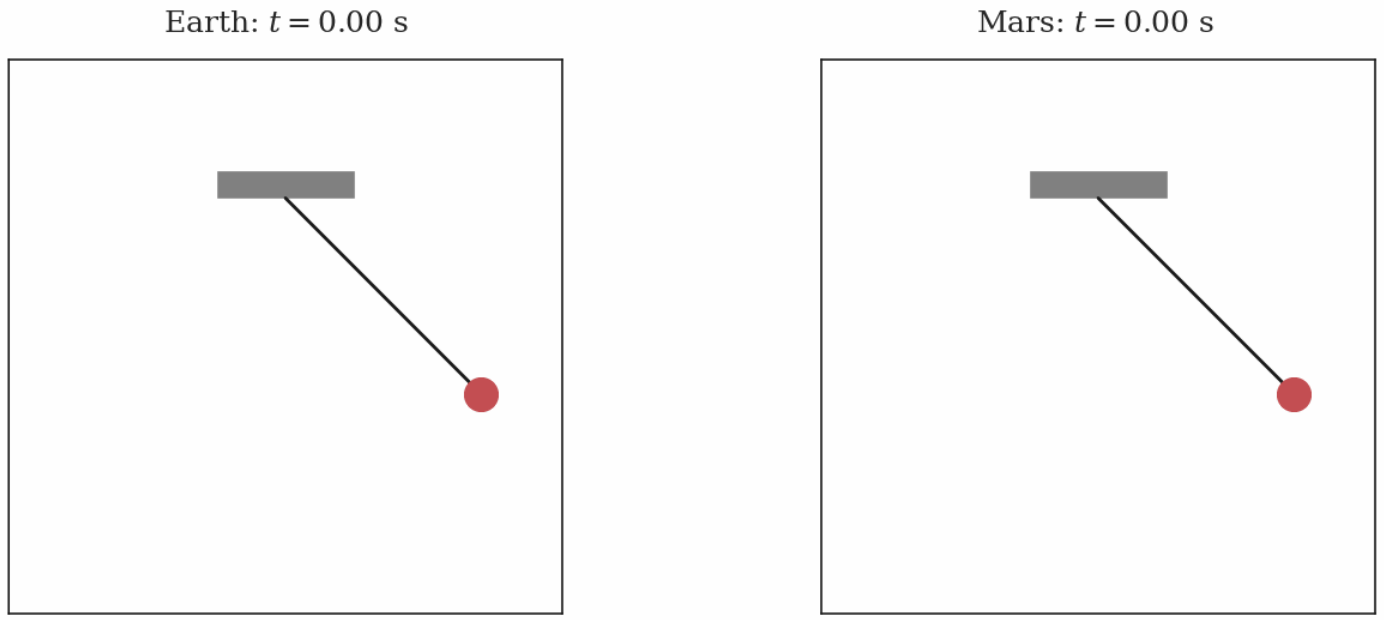
Neural Context Flows for Meta-Learning of Dynamical Systems

Roussel Desmond Nzoyem, David A. W. Barton, Tom Deakin

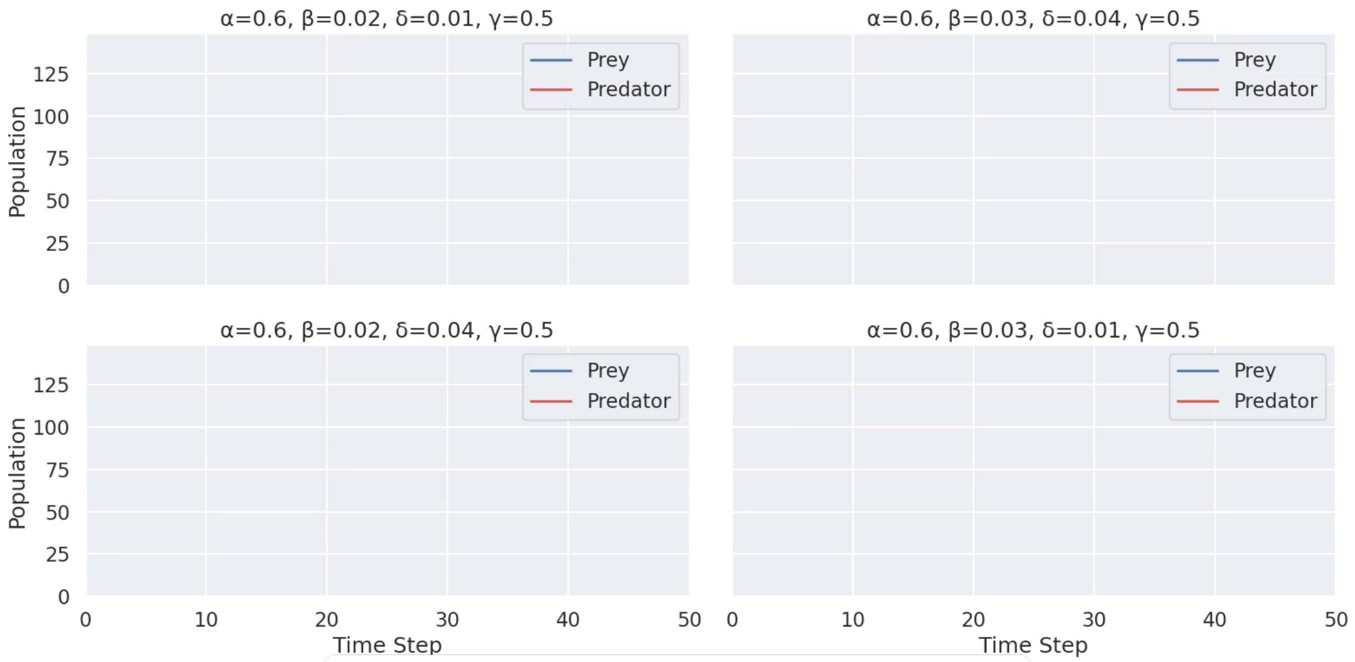
Physical Systems in Nature

Simple Pendulum



$$\begin{cases} \frac{\mathrm{d}\alpha}{\mathrm{d}t} = \omega, \\ \frac{\mathrm{d}\omega}{\mathrm{d}t} = -\frac{g}{L}\sin(\alpha). \end{cases}$$

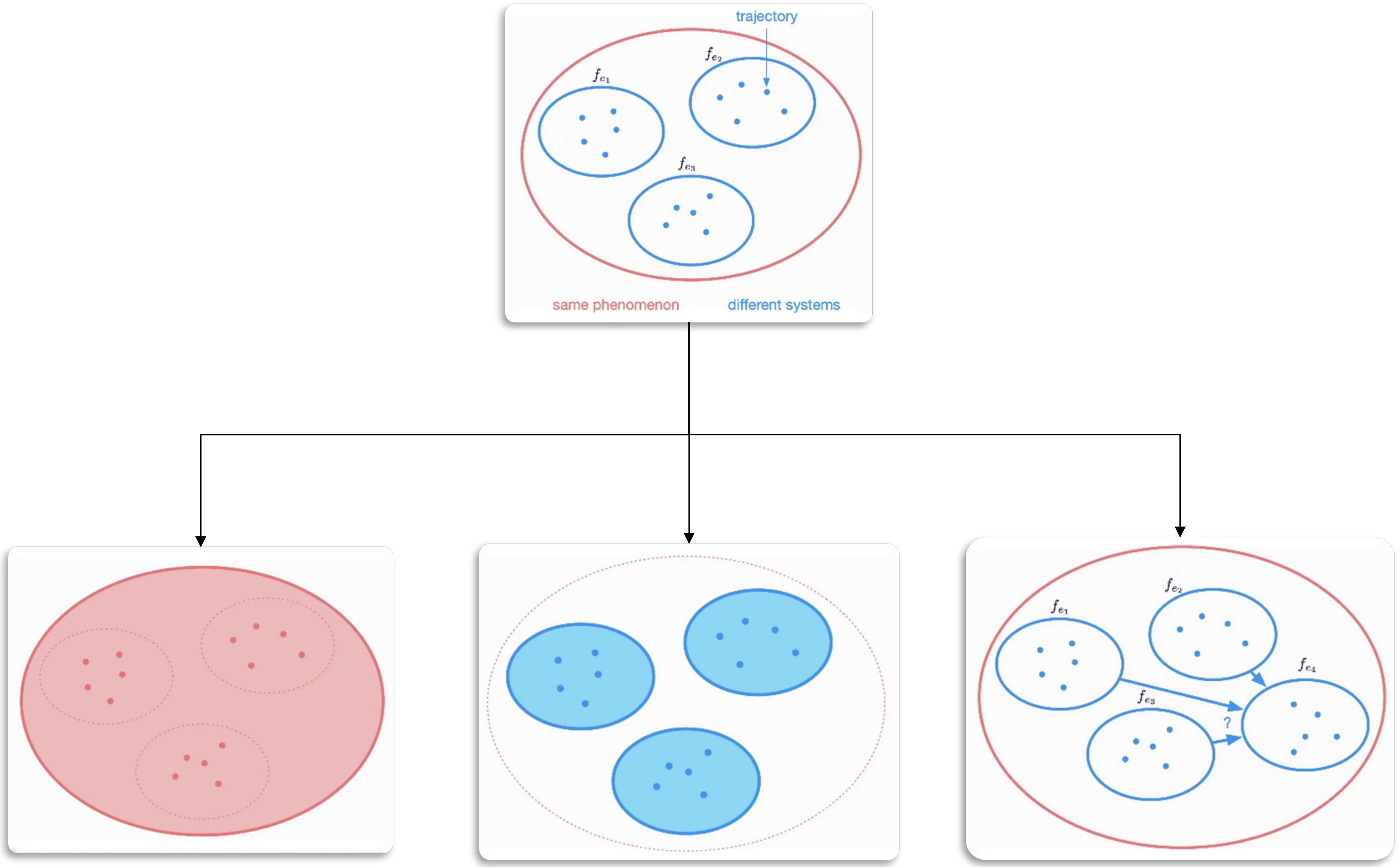
Lotka-Volterra



$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} = \alpha x - \beta xy, \\ \frac{\mathrm{d}y}{\mathrm{d}t} = \delta xy - \gamma y. \end{cases}$$

An **environment** corresponds to an underlying physical parameter value.

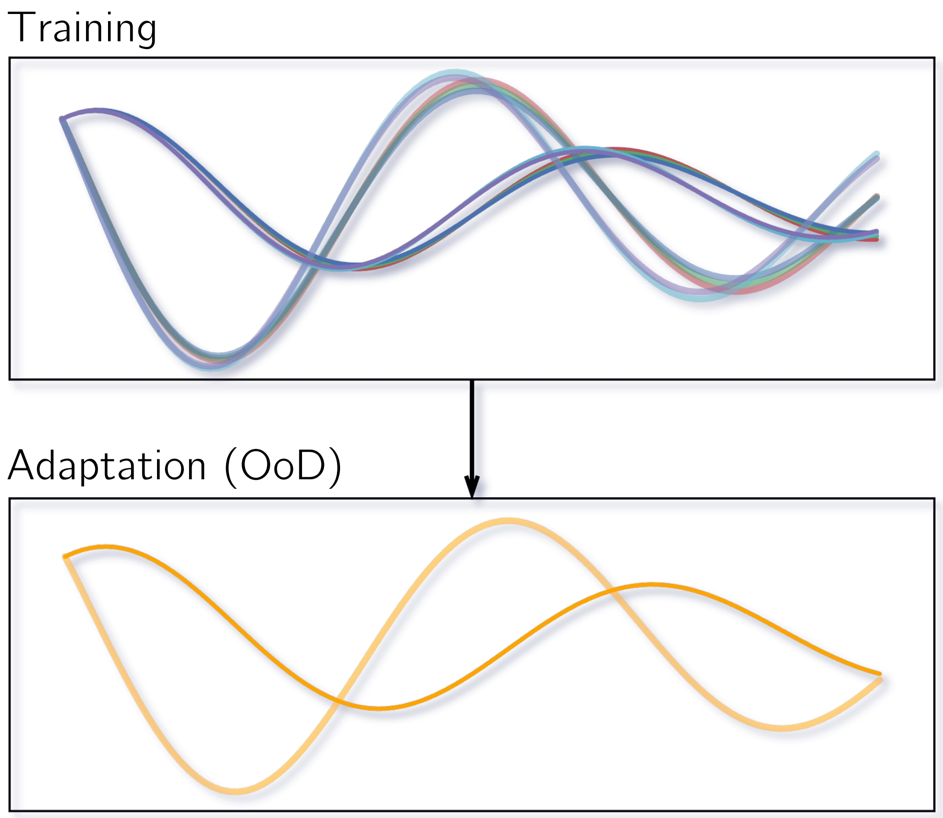
The Generalization Problem



One For All (OFA)

One Per Env (OPE)

- Multi-Task Learning
- Meta-Learning*



Desiderata:
- Few-shot adaptation to new environments
- Quick finetuning for new environments

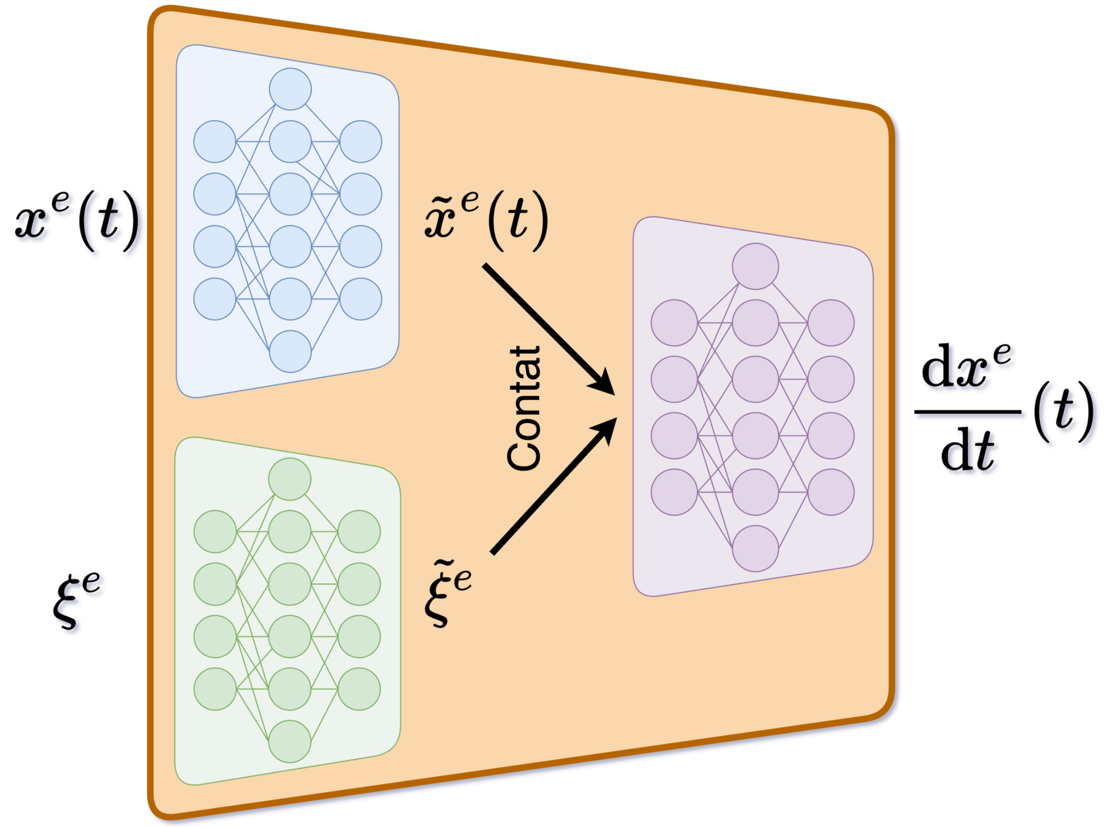
Limitation of existing methods:
- Leveraging information from related environments
- Complex optimization landscapes
- Parameter inefficiency
- Lack of interpretability and uncertainty estimation

Yin et al., "LEADS: Learning Dynamical Systems That Generalize Across Environments", NeurIPS 2021

Our Solution - NCF

1. Introduce learnable context vectors ξ

$$\frac{\mathrm{d}x^e}{\mathrm{d}t} = f_\theta(x^e, \xi^e)$$



3-networks architecture

2. Perform a Taylor expansion about neighboring ξ

$$\frac{\mathrm{d}x^{e,j}}{\mathrm{d}t} = T^k_{f_\theta}(x^{e,j}, \xi^e, \xi^j), \quad \forall j \in \mathrm{P}$$

$$\text{For } k=1, \;\; T^1_{f_\theta}(x^{e,j}, \xi^e, \xi^j) \triangleq f_\theta(x^{e,j}, \xi^j) + \boxed{\nabla_\xi f_\theta(x^{e,j}, \xi^j)(\xi^e - \xi^j)}$$

Why do a Taylor expansion?

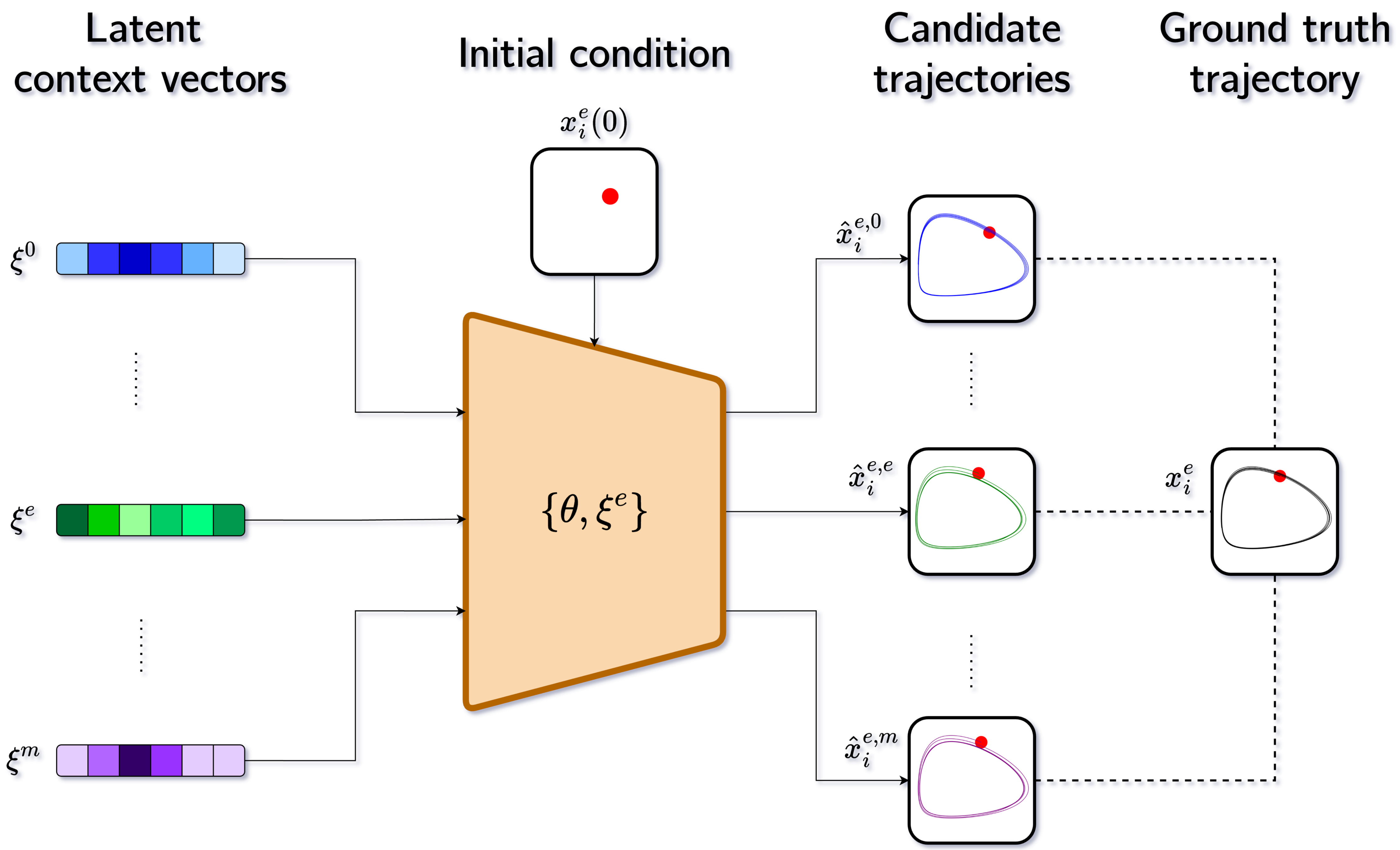
- The vector field is typically differentiable w.r.t. to its physical parameters
- Forces contexts to cluster by **contextual-self modulation**
- Candidate trajectories can be used for uncertainty estimation
- Easy to compute JVPs, including for Taylor orders $k > 1$

Proposition 1 (Second-order Taylor expansion with JVPs). *Assume $f : \mathbb{R}^d \times \mathbb{R}^{d_\xi} \to \mathbb{R}^d$ is C^2 wrt its second argument. Let $x \in \mathbb{R}^d, \xi \in \mathbb{R}^{d_\xi}$, and define $g : \bar{\xi} \mapsto \nabla_\xi f(x, \bar{\xi})(\xi - \bar{\xi})$. The second-order Taylor expansion of f around any $\tilde{\xi} \in \mathbb{R}^\xi$ is then expressed as*

$$f(x,\xi) = f(x,\tilde{\xi}) + \frac{3}{2} g(\tilde{\xi}) + \frac{1}{2} \nabla g(\tilde{\xi})(\xi - \tilde{\xi}) + o(\|\xi - \tilde{\xi}\|^2).$$

Blanke & Lelarge., "Interpretable Meta-Learning of Physical Systems", ICLR 2024

Optimization procedure



During **meta-training**:

- the shared weights θ, and
- the contexts $\{\xi^e\}_{e=1,\ldots,m}$

are optimized in an alternating manner

NCF Variant	Taylor Order	Alternating Minimization Strategy
NCF-t_1	$k = 1$	Ordinary
NCF-t_2	$k = 2$	Proximal

During **adaptation**:

- Only the new context vector is fine-tuned
- Taylor expansion is disabled
- Can be performed in bulk !

Li et al., "Alternating Minimizations Converge to Second-Order Optimal Solutions", ICML 2019

Few-Shot Learning Results

Interpolation

Table 1: Training and adaptation testing MSEs (↓) with OFA, OPE, and NCF-t_1 on the SP problem.

	TRAIN ($\times 10^{-1}$)	ADAPT ($\times 10^{-3}$)
OFA	9.49 ± 0.04	115000 ± 3200
OPE	0.18 ± 0.02	459.0 ± 345.0
NCF	0.10 ± 0.03	0.0356 ± 0.001

Extrapolation

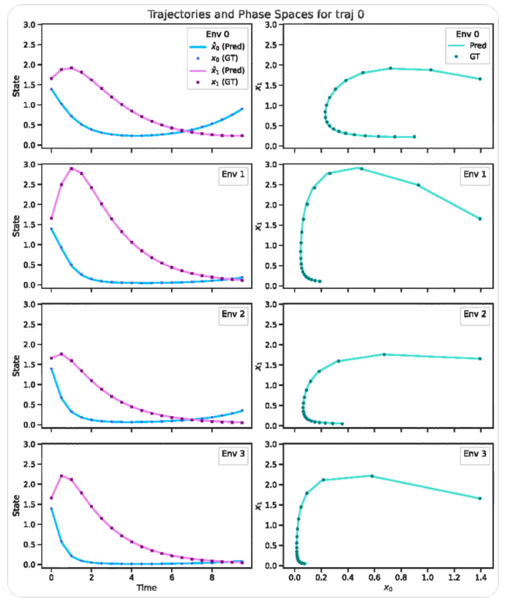
Table 2: In-Domain (InD) and adaptation (OoD) test MSEs (↓) for the LV, GO, SM, BT, GS and NS problems. The best is reported in **bold**. The best of the two NCF variants is shaded in grey.

	LV ($\times 10^{-5}$)			**GO** ($\times 10^{-4}$)		
	#PARAMS	IND	OOD	#PARAMS	IND	OOD
CAVIA	305246	91.0±63.6	120.1±28.3	130711	64.0±14.1	463.4±84.9
CoDA	305793	**1.40±0.13**	2.19±0.78	135390	5.06±0.81	4.22±4.21
NCF-t_1	308240	6.73±0.87	7.92±1.04	131149	40.3±9.1	19.4±1.24
NCF-t_2	308240	1.68±0.32	**1.99±0.31**	131149	**3.33±0.14**	**2.83±0.23**

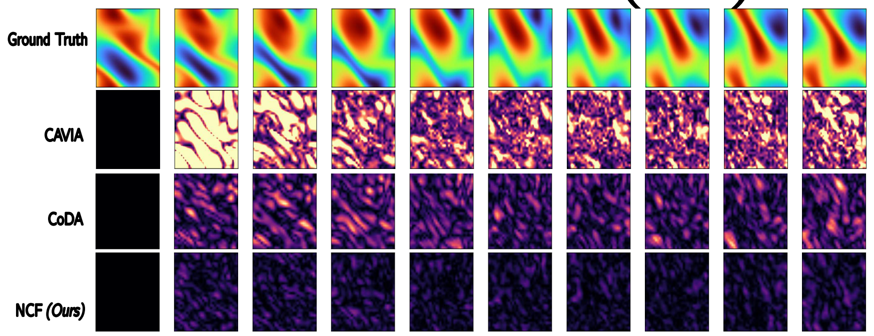
	SM ($\times 10^{-3}$)			**BT** ($\times 10^{-1}$)		
	#PARAMS	IND	OOD	#PARAMS	IND	OOD
CAVIA	50486	979.1±141.2	859.1±70.7	116665	21.93±1.8	22.6±7.22
CoDA	50547	156.0±40.52	8.28±0.29	119679	25.40±9.5	19.47±11.6
NCF-t_1	50000	680.6±320.1	677.2±18.7	117502	21.53±8.9	20.89±12.0
NCF-t_2	50000	**6.42±0.41**	**2.03±0.12**	117502	**3.46±0.09**	**3.77±0.15**

	GS ($\times 10^{-3}$)			**NS** ($\times 10^{-3}$)		
	#PARAMS	IND	OOD	#PARAMS	IND	OOD
CAVIA	618245	69.9±21.2	68.0±4.2	310959	128.1±29.9	126.4±20.7
CoDA	619169	**1.23±0.14**	**0.75±0.65**	309241	7.69±1.14	7.08±0.07
NCF-t_1	610942	7.64±0.70	5.57±0.21	310955	2.98±0.09	2.83±0.06
NCF-t_2	610942	6.15±0.24	3.40±0.51	310955	**2.92±0.08**	**2.79±0.09**

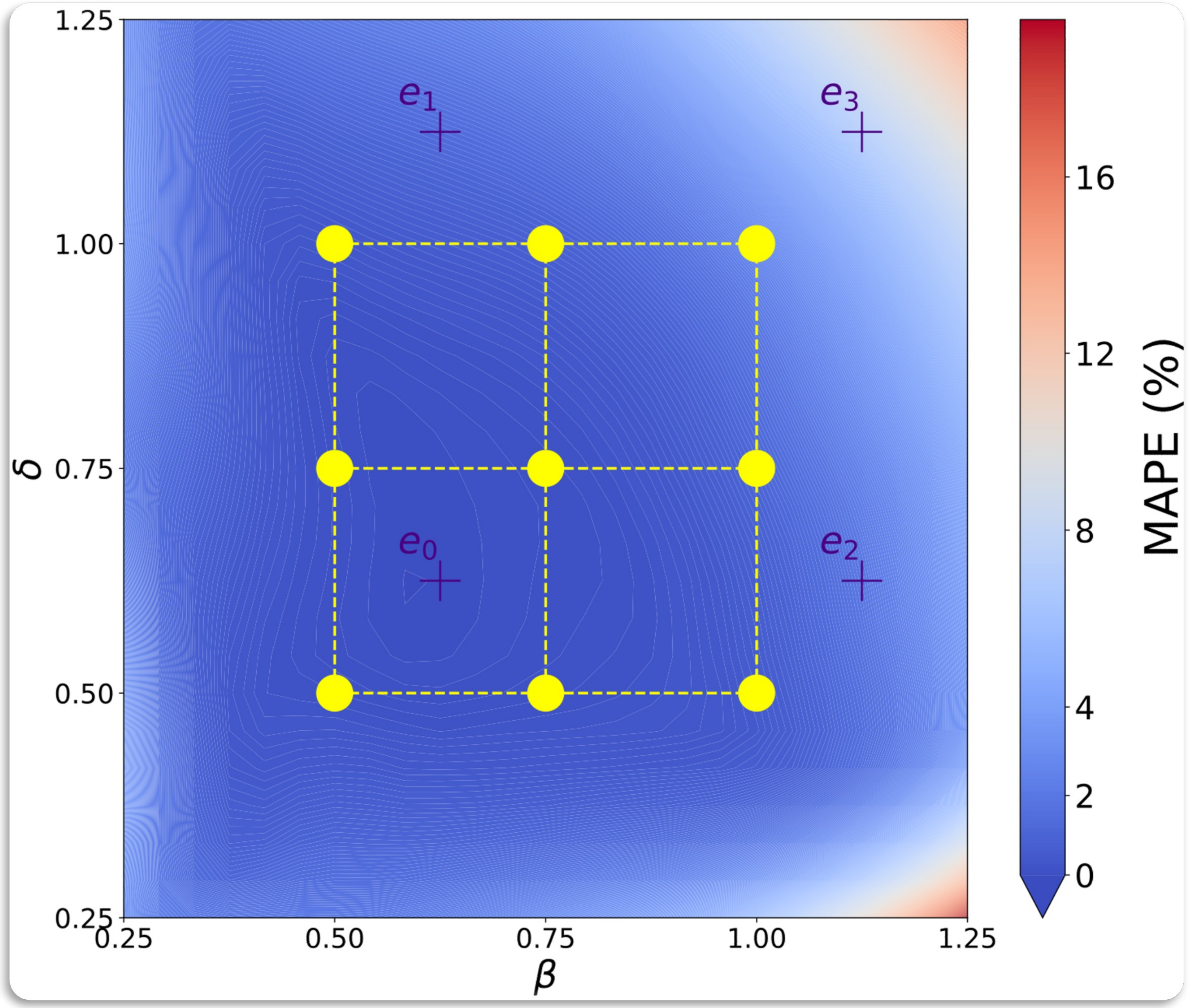
Lotka-Volterra (LV)



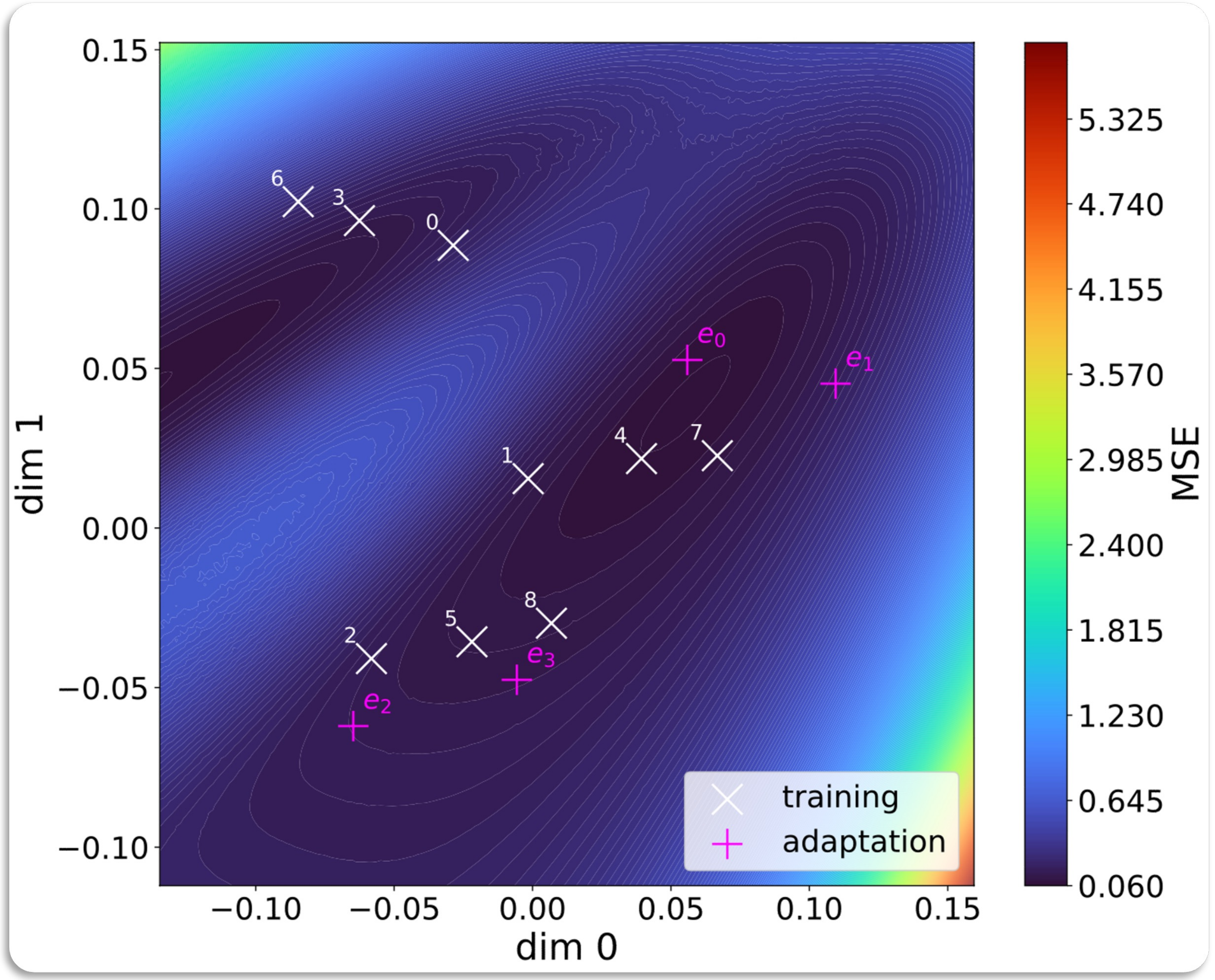
Navier-Stokes (NS)



Analyzing the Landscapes

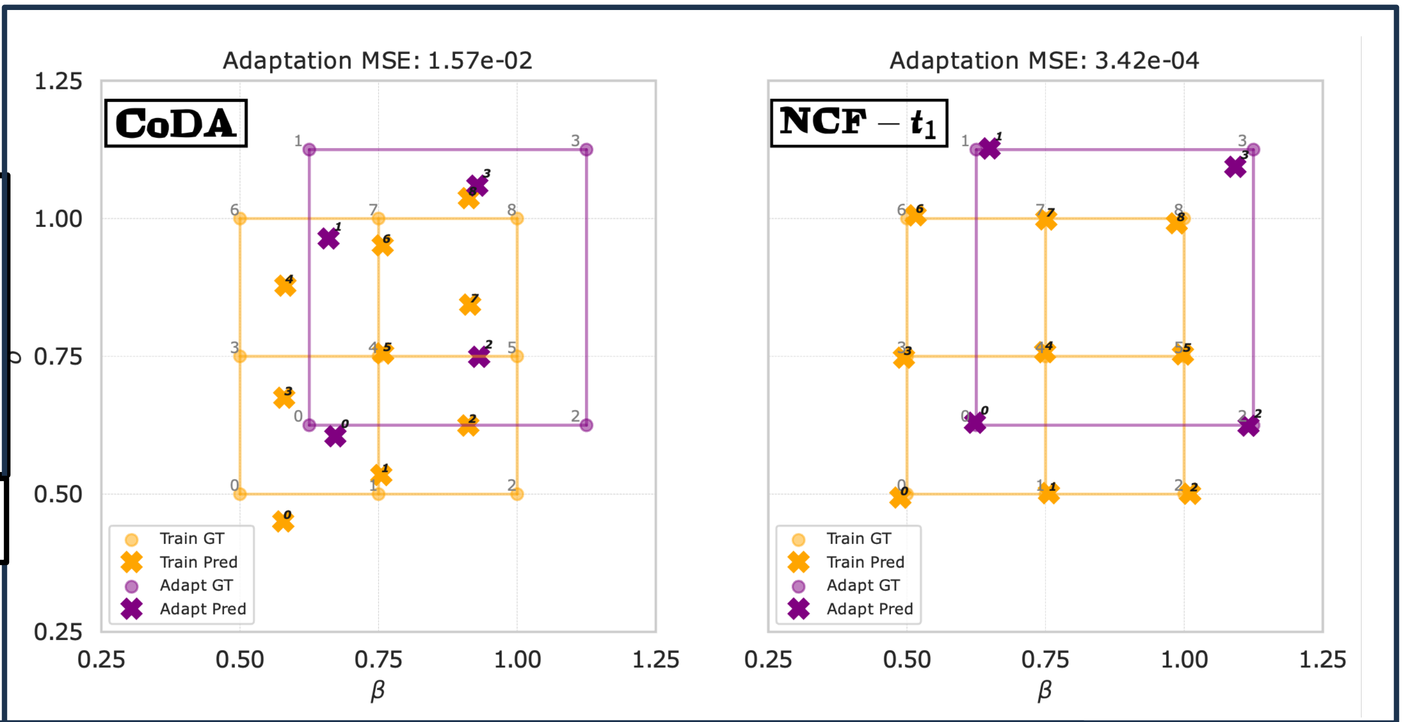


Grid-Wise Adaptation



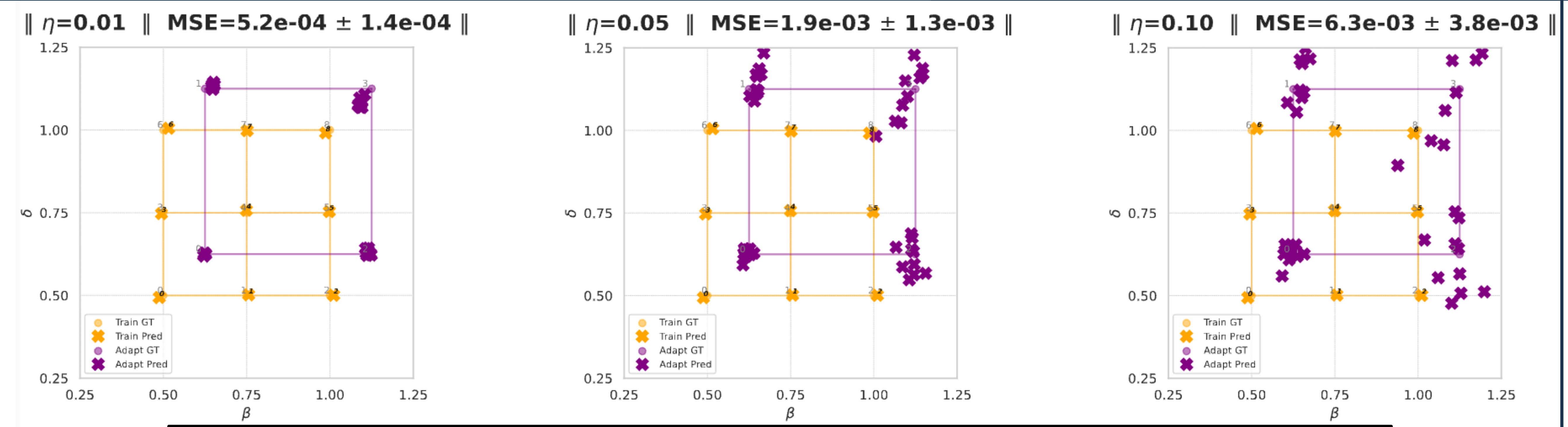
Loss Landscape

Interpretability of Context Vectors



Proposition 2 (Identifiability of affine systems). *Assume $d_\xi \geq d_c$, that P is full-rank, and that f_{true} is differentiable in its second argument. In the limit of zero training loss in Eq. (10), f_θ trained with first-order Taylor expansion and the Random-All pool-filling strategy[4] is affine on an open region of $\mathbb{R}^{d_\xi}$. Furthermore, there exists $Q \in \mathbb{R}^{d_c \times d_\xi}$ and $q \in \mathbb{R}^{d_c}$ such that for any meta-trained $\xi \in \{\xi^e\}_{e=1}^m$ and its corresponding underlying parameter $c \in \{c^e\}_{e=1}^m$, we have $c = Q\xi + q$.*

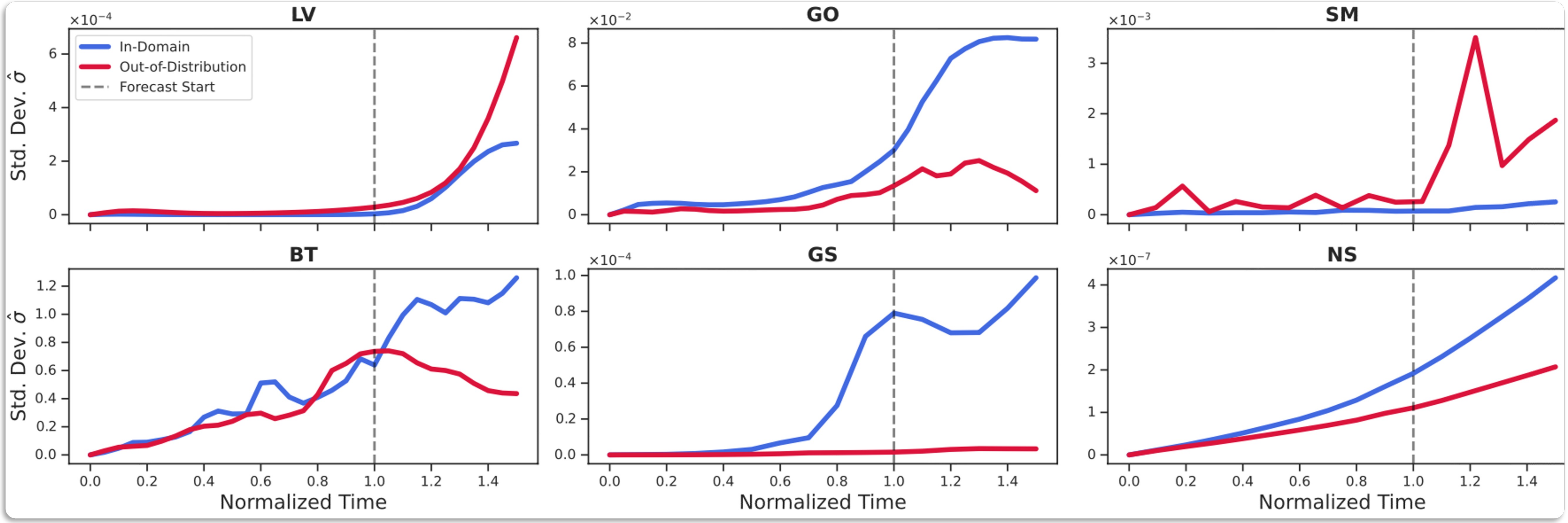
Parameters of affine systems are recoverable up to an affine transform.



Affine system identification with NCF is robust to noise.

Uncertainty Estimation

Standard deviation across candidate trajectories grows with time !



Neural Context Flow in a Nutshell

We propose a powerful framework for meta-learning dynamical systems:

1. Achieves SoTA results on common datasets
2. Parallelizable and scalable
3. Interpretable and robust to noise
4. Provides uncertainty estimation

To improve reproducibility, we propose Gen-Dynamics

https://github.com/ddrous/gen-dynamics

README   GPL-3.0 license

Gen Dynamics

A collection of datasets for learning generalisable dynamical systems

Learning dynamical systems that can generalize to various parameter changes in their underlying ODEs or PDEs is a significant but challenging task. In recent years, several methods have been proposed to learn such systems. This repository contains a collection of datasets that can be used to benchmark these methods.

We aim to provide a consistent interface to access each dataset. The datasets are made up of 4 splits each:

1. `train` : For In-Domain meta-learning
2. `test` : For In-Domain meta-testing
3. `ood_train` : For Out-of-Distribution adaptation to new environments
4. `ood_test` : For Out-of-Distribution testing of the adaptation to new environments

Thank You !

Paper: arxiv.org/abs/2405.02154

OpenReview

Code:

- JAX : github.com/ddrous/ncflow
- PyTorch : github.com/ddrous/ncflow-torch
- Gen-Dynamics: github.com/ddrous/gen-dynamics

Contact: rd.nzoyemngueguin@bristol.ac.uk

NEURAL CONTEXT FLOWS FOR META-LEARNING OF DYNAMICAL SYSTEMS

Roussel Desmond Nzoyem
School of Computer Science
University of Bristol
rd.nzoyemngueguin@bristol.ac.uk

David A.W. Barton
School of Engineering Mathematics and Technology
University of Bristol
david.barton@bristol.ac.uk

Tom Deakin
School of Computer Science
University of Bristol
tom.deakin@bristol.ac.uk

ABSTRACT

Neural Ordinary Differential Equations (NODEs) often struggle to adapt to new dynamic behaviors caused by parameter changes in the underlying physical system, even when these dynamics are similar to previously observed behaviors. This problem becomes more challenging when the changing parameters are unobserved, meaning their value or influence cannot be directly measured when collecting data. To address this issue, we introduce Neural Context Flow (NCF), a robust and interpretable Meta-Learning framework that includes uncertainty estimation. NCF uses Taylor expansion to enable contextual self-modulation, allowing context vectors to influence dynamics from other domains while also modulating themselves. After establishing theoretical guarantees, we empirically test NCF and compare it to related adaptation methods. Our results show that NCF achieves state-of-the-art Out-of-Distribution performance on 5 out of 6 linear and non-linear benchmark problems. Through extensive experiments, we explore the flexible model architecture of NCF and the encoded representations within the learned context vectors. Our findings highlight the potential implications of NCF for foundational models in the physical sciences, offering a promising approach to improving the adaptability and generalization of NODEs in various scientific applications.