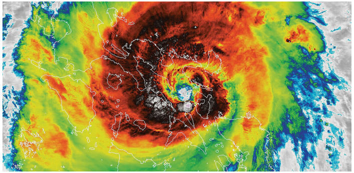
Generalizing to New Physical Systems via Context-Informed Dynamics Model

ICML 2022

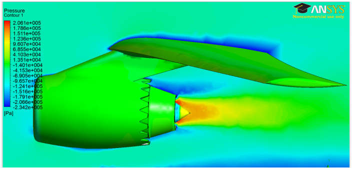
July 17th–23rd, 2022

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

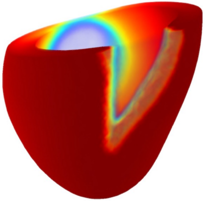
[*] *Equal Contribution*, [1]*Sorbonne Université - MLIA ISIR*, [2]*Criteo AI Lab*, [3]*Université de Rouen - LITIS*



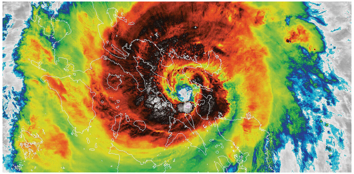
Weather forecasting



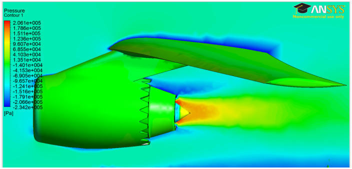
Airplane design



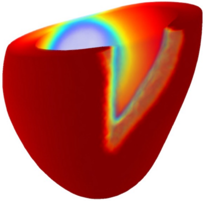
Heart dynamics



Weather forecasting



Airplane design

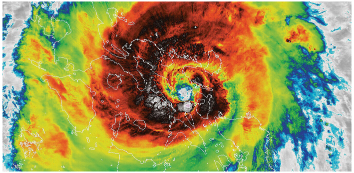


Heart dynamics

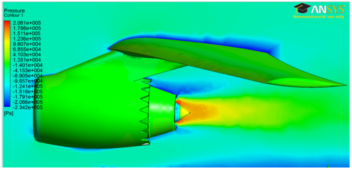
Modelling dynamics from data with NNs

➜ Strong + flexible alternative to *physical models*.

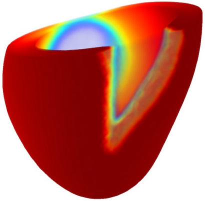
SCIENCES SORBONNE UNIVERSITÉ



Weather forecasting



Airplane design



Heart dynamics

Modelling dynamics from data with NNs

→ Strong + flexible alternative to *physical models*.

→ Successfully applied to various problems (Li et al., 2021; Sirignano and Spiliopoulos, 2018; de Bézenac et al., 2018).

Li et al., *Fourier Neural Operator for Parametric Partial Differential Equations. ICLR, 2021*
Sirignano and Spiliopoulos, *DGM: A deep learning algorithm for solving partial differential equations. Journal of Computational Physics, 2018*
de Bézenac et al., *Deep Learning for Physical Processes: Incorporating Prior Scientific Knowledge. ICLR, 2018*

NNs and OOD generalization

→ NNs generalize poorly out-of-distribution.

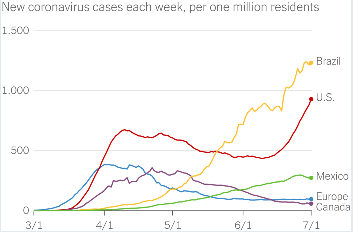
SCIENCES SORBONNE UNIVERSITÉ

NNs and OOD generalization

- ➜ NNs generalize poorly out-of-distribution.
- ➜ Limitation for real-world dynamics models, e.g. when modelling:

NNs and OOD generalization

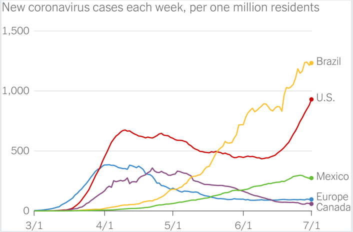
- ➜ NNs generalize poorly out-of-distribution.
- ➜ Limitation for real-world dynamics models, e.g. when modelling:



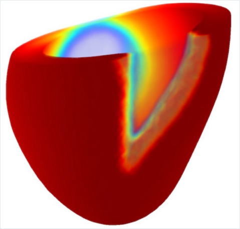
Disease diffusion across countries

NNs and OOD generalization

- ➜ NNs generalize poorly out-of-distribution.
- ➜ Limitation for real-world dynamics models, e.g. when modelling:



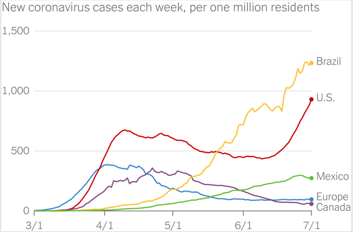
Disease diffusion across countries



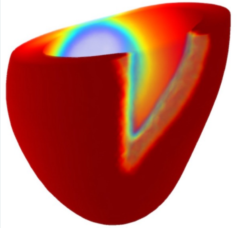
Heart dynamics across patients

NNs and OOD generalization

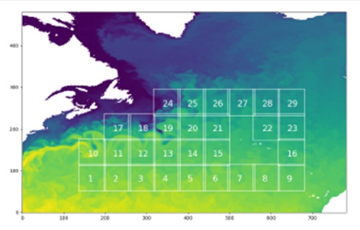
- ➜ NNs generalize poorly out-of-distribution.
- ➜ Limitation for real-world dynamics models, e.g. when modelling:



Disease diffusion across countries



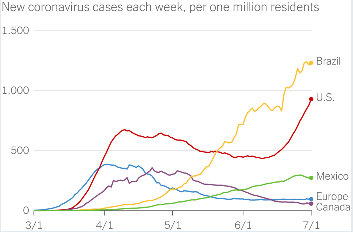
Heart dynamics across patients



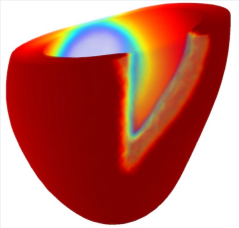
Sea surface temperature across spatial regions

NNs and OOD generalization

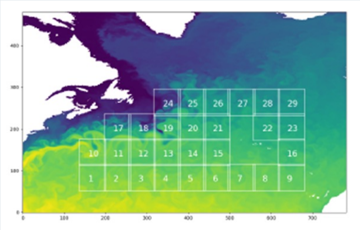
- ➜ NNs generalize poorly out-of-distribution.
- ➜ Limitation for real-world dynamics models, e.g. when modelling:



Disease diffusion across countries



Heart dynamics across patients



Sea surface temperature across spatial regions

➜ Context-Informed Dynamics Adaptation (CoDA)
➥ one of the first principled solution to this open generalization problem.

SCIENCES SORBONNE UNIVERSITÉ

Notation and objective

Physical systems driven by *unknown* differential equations:

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

Notation and objective

Physical systems driven by *unknown* differential equations:

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

➜ $x(t)$: state value at t.

SCIENCES SORBONNE UNIVERSITÉ

Notation and objective

Physical systems driven by *unknown* differential equations:

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

- ➜ $x(t)$: state value at t.
- ➜ f: *unknown* dynamics.

SCIENCES SORBONNE UNIVERSITÉ

Notation and objective

Physical systems driven by *unknown* differential equations:

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

- $x(t)$: state value at t.
- f: *unknown* dynamics.
  - defined by a *physical context*: system parameters, external forces...

Notation and objective

Physical systems driven by *unknown* differential equations:

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

- ➜ $x(t)$: state value at t.
- ➜ f: *unknown* dynamics.
  - ➥ defined by a *physical context*: system parameters, external forces...

Goal: Learn dynamics across contexts with a *neural dynamics model* g_θ.

Notation and objective

Physical systems driven by *unknown* differential equations:

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

- ➜ $x(t)$: state value at t.
- ➜ f: *unknown* dynamics.
  - ➥ defined by a *physical context*: system parameters, external forces...

Goal: Learn dynamics across contexts with a *neural dynamics model* g_θ.

Multi-environment learning problem

Formalization

Notation and objective

Physical systems driven by *unknown* differential equations:

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

- ➜ $x(t)$: state value at t.
- ➜ f: *unknown* dynamics.
  - ➥ defined by a *physical context*: system parameters, external forces...

Goal: Learn dynamics across contexts with a *neural dynamics model* g_θ.

Multi-environment learning problem

- ➜ Environment e:

Formalization

Notation and objective

Physical systems driven by *unknown* differential equations:

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

- ➜ $x(t)$: state value at t.
- ➜ f: *unknown* dynamics.
  - ➥ defined by a *physical context*: system parameters, external forces...

Goal: Learn dynamics across contexts with a *neural dynamics model* g_θ.

Multi-environment learning problem

- ➜ Environment e:
  - ➥ physical context.

SCIENCES SORBONNE UNIVERSITÉ

Notation and objective

Physical systems driven by *unknown* differential equations:

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

- ➜ $x(t)$: state value at t.
- ➜ f: *unknown* dynamics.
  - ➥ defined by a *physical context*: system parameters, external forces...

Goal: Learn dynamics across contexts with a *neural dynamics model* g_θ.

Multi-environment learning problem

- ➜ Environment e:
  - ➥ physical context.
  - ➥ several observed trajectories of f^e.

SCIENCES SORBONNE UNIVERSITÉ

Notation and objective

Physical systems driven by *unknown* differential equations:

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

- $x(t)$: state value at t.
- f: *unknown* dynamics.
  - defined by a *physical context*: system parameters, external forces...

Goal: Learn dynamics across contexts with a *neural dynamics model* g_θ.

Multi-environment learning problem

- Environment e:
  - physical context.
  - several observed trajectories of f^e.
- **Training**: environments $\mathcal{E}_{\text{tr}}$ with *reasonable data*.

Formalization

Notation and objective

Physical systems driven by *unknown* differential equations:

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

- ➜ $x(t)$: state value at t.
- ➜ f: *unknown* dynamics.
  - ➥ defined by a *physical context*: system parameters, external forces...

Goal: Learn dynamics across contexts with a *neural dynamics model* g_θ.

Multi-environment learning problem

- ➜ Environment e:
  - ➥ physical context.
  - ➥ several observed trajectories of f^e.
- ➜ **Training**: environments $\mathcal{E}_{\text{tr}}$ with *reasonable data*.
- ➜ **Adaptation**: generalize to new environments $\mathcal{E}_{\text{ad}}$ with *few data*.

SCIENCES SORBONNE UNIVERSITÉ

CoDA learns

CoDA learns

➜ shared parameters across environments, θ^c.

CoDA learns

- ➜ shared parameters across environments, θ^c.
- ➜ environment-specific parameters, $\delta\theta^e$.

CoDA learns

- ➜ shared parameters across environments, θ^c.
- ➜ environment-specific parameters, $\delta\theta^e$.
- ➜ $\theta^e \triangleq \theta^c + \delta\theta^e$.

CoDA learns

- ➜ shared parameters across environments, θ^c.
- ➜ environment-specific parameters, $\delta\theta^e$.
- ➜ $\theta^e \triangleq \theta^c + \delta\theta^e$.

Under two constraints:

CoDA learns

- ➜ shared parameters across environments, θ^c.
- ➜ environment-specific parameters, $\delta\theta^e$.
- ➜ $\theta^e \triangleq \theta^c + \delta\theta^e$.

Under two constraints:

Locality.

CoDA learns

- ➜ shared parameters across environments, θ^c.
- ➜ environment-specific parameters, $\delta\theta^e$.
- ➜ $\theta^e \triangleq \theta^c + \delta\theta^e$.

Under two constraints:

Locality.

Fast adaptation to new systems.

CoDA learns

- ➜ shared parameters across environments, θ^c.
- ➜ environment-specific parameters, $\delta\theta^e$.
- ➜ $\theta^e \triangleq \theta^c + \delta\theta^e$.

Under two constraints:

Locality.

Fast adaptation to new systems.

Low-rank adaptation.

SCIENCES SORBONNE UNIVERSITÉ

CoDA learns

- ➜ shared parameters across environments, θ^c.
- ➜ environment-specific parameters, $\delta\theta^e$.
- ➜ $\theta^e \triangleq \theta^c + \delta\theta^e$.

Under two constraints:

Locality.	Fast adaptation to new systems.
Low-rank adaptation.	Low-dimensionality of the context.

SCIENCES SORBONNE UNIVERSITÉ

Constrained optimization problem

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

SCIENCES SORBONNE UNIVERSITÉ

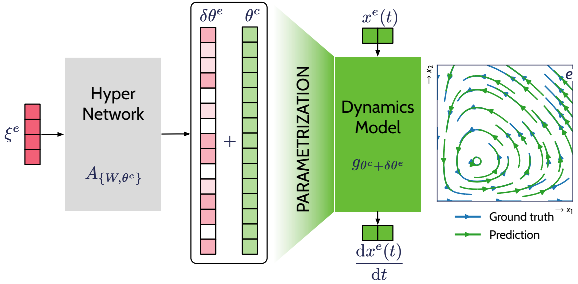
Constrained optimization problem

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

➜ Fast adaptation.

θ^e generated via a linear **hypernetwork** $A_{\{W,\theta^c\}}$:

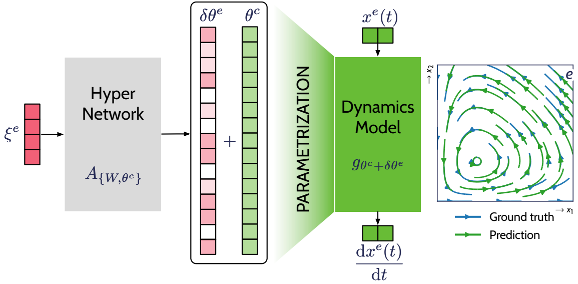
$$\theta^e \triangleq A_{\{W,\theta^c\}}(\xi^e) = \theta^c + W\xi^e$$



θ^e generated via a linear **hypernetwork** $A_{\{W,\theta^c\}}$:

$$\theta^e \triangleq A_{\{W,\theta^c\}}(\xi^e) = \theta^c + W\xi^e$$

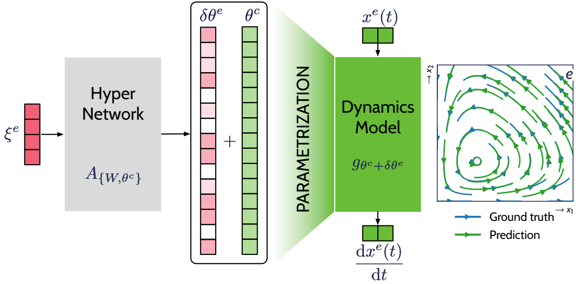
➜ ξ^e: context vector - low-dimensional, *environment-specific*



θ^e generated via a linear **hypernetwork** $A_{\{W,\theta^c\}}$:

$$\theta^e \triangleq A_{\{W,\theta^c\}}(\xi^e) = \theta^c + W\xi^e$$

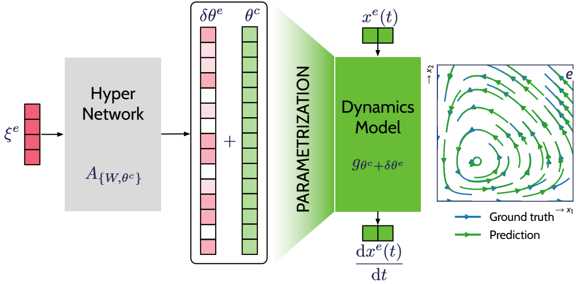
- ➜ ξ^e: context vector - low-dimensional, *environment-specific*
- ➜ Rows of W: *shared* adaptation directions



θ^e generated via a linear **hypernetwork** $A_{\{W,\theta^c\}}$:

$$\theta^e \triangleq A_{\{W,\theta^c\}}(\xi^e) = \theta^c + W\xi^e$$

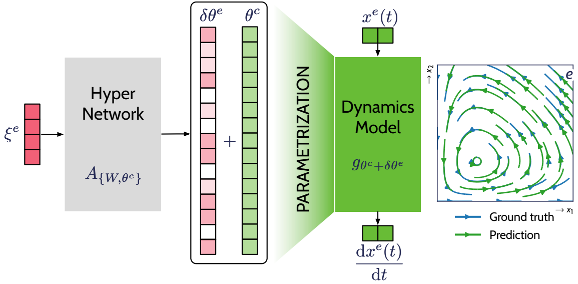
- ➜ ξ^e: context vector - low-dimensional, *environment-specific*
- ➜ Rows of W: *shared* adaptation directions
  - ➥ Fixed low-dimensional adaptation subspace $\mathcal{W} \triangleq \mathrm{Span}(W_1, \cdots, W_{\dim(\xi)})$



θ^e generated via a linear **hypernetwork** $A_{\{W,\theta^c\}}$:

$$\theta^e \triangleq A_{\{W,\theta^c\}}(\xi^e) = \theta^c + W\xi^e$$

- ➜ ξ^e: context vector - low-dimensional, *environment-specific*
- ➜ Rows of W: *shared* adaptation directions
  - ➥ Fixed low-dimensional adaptation subspace $\mathcal{W} \triangleq \mathrm{Span}(W_1, \cdots, W_{\dim(\xi)})$

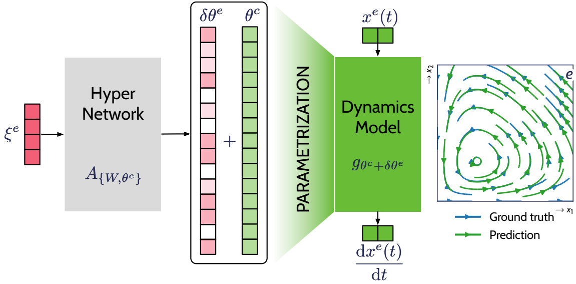


➜ Adaptation is parameter efficient.

θ^e generated via a linear **hypernetwork** $A_{\{W,\theta^c\}}$:

$$\theta^e \triangleq A_{\{W,\theta^c\}}(\xi^e) = \theta^c + W\xi^e$$

- ➜ ξ^e: context vector - low-dimensional, *environment-specific*
- ➜ Rows of W: *shared* adaptation directions
  - ➥ Fixed low-dimensional adaptation subspace $\mathcal{W} \triangleq \mathrm{Span}(W_1, \cdots, W_{\dim(\xi)})$



- ➜ Adaptation is parameter efficient.
- ➜ Experimentally sample-efficient.

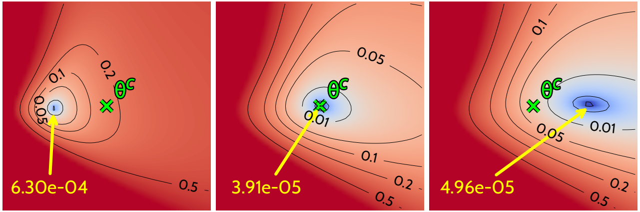


Figure 1: CoDA's projected loss landscape onto $\mathcal{W}$ for 3 *Lotka-Volterra* systems.

Loss landscape

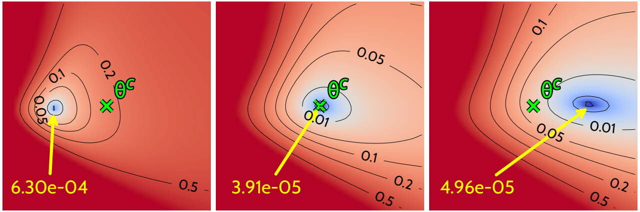


Figure 1: CoDA's projected loss landscape onto $\mathcal{W}$ for 3 *Lotka-Volterra* systems.

Low-rank

➜ Adaptation subspace $\mathcal{W}$ contains optima ($\longrightarrow$) with low loss.

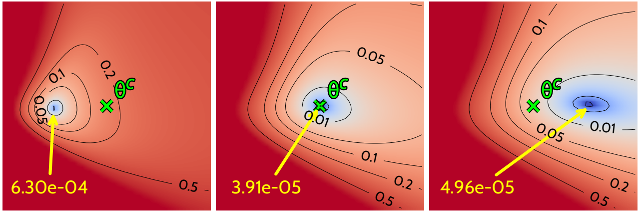


Figure 1: CoDA's projected loss landscape onto $\mathcal{W}$ for 3 *Lotka-Volterra* systems.

Low-rank

➜ Adaptation subspace $\mathcal{W}$ contains optima ($\longrightarrow$) with low loss.

Locality

➜ Proximity of optima ($\longrightarrow$) to θ^c (×).

Experimental setting

➜ Datasets: ODEs + PDEs with *unknown* parameters that vary across systems.

Table 1: MSE on new test trajectories (↓). Best in **bold**; Second <u>underlined</u>.

	Lotka-Volterra $\times 10^{-5}$	*Glycolitic-Oscillator* $\times 10^{-4}$	*Gray-Scott* $\times 10^{-3}$	*Navier-Stokes* $\times 10^{-4}$

Quantitative results

Experimental setting

- ➜ Datasets: ODEs + PDEs with *unknown* parameters that vary across systems.
- ➜ Dynamics-aware baselines based on meta-learning; multi-task learning.

Table 1: MSE on new test trajectories (↓). Best in **bold**; Second <u>underlined</u>.

	Method	*Lotka-Volterra* $\times 10^{-5}$	*Glycolitic-Oscillator* $\times 10^{-4}$	*Gray-Scott* $\times 10^{-3}$	*Navier-Stokes* $\times 10^{-4}$
GBML	MAML				
	ANIL				
	Meta-SGD				
MTL	LEADS				
Contextual	CAVIA-FiLM				
	CAVIA-Concat				
	CoDA-ℓ_2				
	CoDA-ℓ_1				

SCIENCES SORBONNE UNIVERSITÉ

Experimental setting

- Datasets: ODEs + PDEs with *unknown* parameters that vary across systems.
- Dynamics-aware baselines based on meta-learning; multi-task learning.
- Evaluation:
  - In-Domain ($\mathcal{E}_{\text{tr}}$).

Table 1: MSE on new test trajectories (↓). Best in **bold**; Second <u>underlined</u>.

	Method	*Lotka-Volterra* $\times 10^{-5}$	*Glycolitic-Oscillator* $\times 10^{-4}$	*Gray-Scott* $\times 10^{-3}$	*Navier-Stokes* $\times 10^{-4}$
		In-domain	In-domain	In-domain	In-domain
GBML	MAML	60.3±1.3	57.3±2.1	3.67±0.53	68.0±8.0
GBML	ANIL	381±76	74.5±11.5	5.01±0.80	61.7±4.3
GBML	Meta-SGD	32.7±12.6	42.3±6.9	2.85±0.54	53.9±28.1
MTL	LEADS	3.70±0.27	31.4±3.3	2.90±0.76	14.0±1.55
Contextual	CAVIA-FiLM	4.38±1.15	4.44±1.46	2.81±1.15	23.2±12.1
Contextual	CAVIA-Concat	2.43±0.66	5.09±0.35	2.67±0.48	25.5±6.31
Contextual	**CoDA-ℓ_2**	<u>1.52±0.08</u>	<u>2.45±0.38</u>	<u>1.01±0.15</u>	<u>9.40±1.13</u>
Contextual	**CoDA-ℓ_1**	**1.35±0.22**	**2.20±0.26**	**0.90±0.057**	**8.35±1.71**

SCIENCES SORBONNE UNIVERSITÉ

Experimental setting

- Datasets: ODEs + PDEs with *unknown* parameters that vary across systems.
- Dynamics-aware baselines based on meta-learning; multi-task learning.
- Evaluation:
  - In-Domain ($\mathcal{E}_{\text{tr}}$).
  - 1-shot Adaptation ($\mathcal{E}_{\text{ad}}$).

Table 1: MSE on new test trajectories (↓). Best in **bold**; Second underlined.

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

Take-home messages

→ New SoTA approach to handle OOD generalization in dynamical systems.

Conclusion

Take-home messages

➜ New SoTA approach to handle OOD generalization in dynamical systems.

➜ Fast, parameter-efficient and sample-efficient adaptation.

SCIENCES SORBONNE UNIVERSITÉ

Take-home messages

- ➜ New SoTA approach to handle OOD generalization in dynamical systems.
- ➜ Fast, parameter-efficient and sample-efficient adaptation.

Paper `arxiv.org/abs/2202.01889`
Code `github.com/yuan-yin/CoDA`
Contact `{matthieu.kirchmeyer,yuan.yin}@isir.upmc.fr`

SCIENCES SORBONNE UNIVERSITÉ

Take-home messages

- ➜ New SoTA approach to handle OOD generalization in dynamical systems.
- ➜ Fast, parameter-efficient and sample-efficient adaptation.

Paper `arxiv.org/abs/2202.01889`
Code `github.com/yuan-yin/CoDA`
Contact `{matthieu.kirchmeyer,yuan.yin}@isir.upmc.fr`

Check out our poster: Hall E #313 19/07 - 5:30 p.m. — 7:30 p.m !