Deep augmented physical models: application to computer vision and reinforcement learning

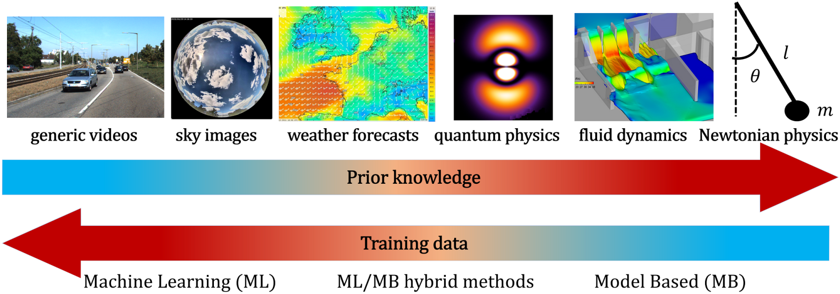
Vincent Le Guen

GDR Mascott-Num workshop on Physics-informed learning
05/12/2023

Outline

Modelling approaches for dynamics forecasting

- **Model-Based (MB) approaches**: requires a deep understanding of the underlying phenomenon, e.g. formalized by ODE/PDE
- **Machine Learning (ML) / Deep Learning**: more agnostic, now state-of-the-art for several tasks, e.g. ConvLSTM [Shi et al., 2015], NeuralODE [Chen et al., 2018]
- **Hybrid ML/MB**: cooperation between physical models and data, historically data assimilation [Corpetti et al., 2009], new hot topic in ML

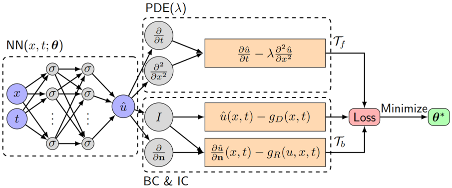


Incorporating full physical knowledge in ML

Hybrid models combining MB and ML (gray box)

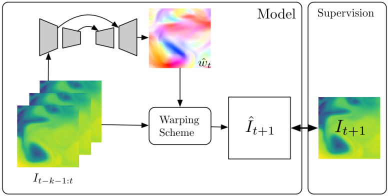
- A long-standing topic [Rico-Martinez et al., 1994]
- 2 main categories of methods:

Loss function regularization



Physics-informed neural networks [Raissi, 2018]

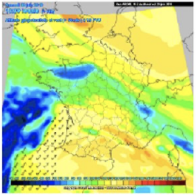
Constraints in deep architectures

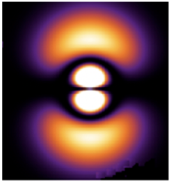


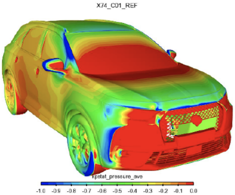
Advection-diffusion model [de Bezenac et al., 2018]

Focus: simplified physical models

Physical models: often approximations of real-world dynamics







- A complete description of a complex natural phenomenon is out of reach, e.g. climate, earth modelling
- Approximations are made to make the numerical resolution tractable, e.g. reduced-order models, resolution on coarse meshes

Outline

Augmenting simplified physical models

- Let's consider the dynamical system $\frac{\mathrm{d}X_t}{\mathrm{d}t} = F(X_t)$, F unknown
- We often have a rough physical prior model F_p, with unknown params θ_p
- We consider a linear augmentation: $\boxed{F = F_p + F_a}$

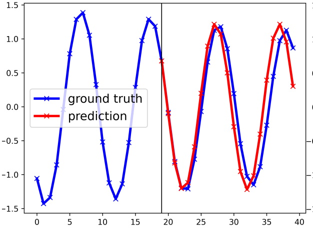
Motivation: principled ML/MB decomposition

Issue: unconstrained decomposition $\frac{\mathrm{d}X_t}{\mathrm{d}t} = (F_p + F_a)(X_t)$ often ill-posed

- possibly an infinite number of solutions
- wrong identification of the physical parameters θ_p

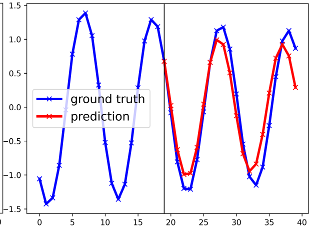
Damped pendulum: $\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} + \omega_0^2 \sin\theta + \lambda \frac{\mathrm{d}\theta}{\mathrm{d}t} = 0$

MSE=1.1 10^{-1}, Err T_0=5.7%



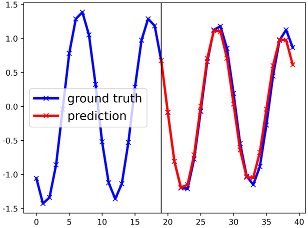
Simplified physical model

MSE=8.2 10^{-2}, Err T_0=12.4%



Unconstrained augmentation

MSE=1.5 10^{-2}, Err T_0=1.4%



APHYNITY framework

APHYNITY [ICLR'21]

Augmenting PHYsical models for ideNtIfying and forecasTing complex dYnamics

- The physical model F_p should explain the dynamics as much as possible
- The complete model $F = F_p + F_a$ should perfectly explain the dynamics

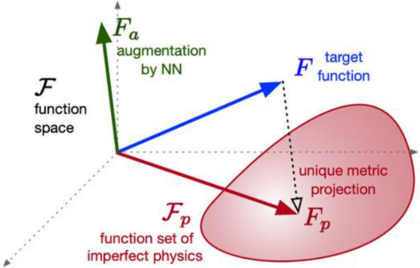
APHYNITY objective:

$\mathcal{F}$: normed vector space

$\mathcal{F}_p \subset \mathcal{F}$: space of physical models

$$\min_{F_p \in \mathcal{F}_p, F_a \in \mathcal{F}} \quad \|F_a\| \quad \text{subject to}$$

$$\forall X \in \mathcal{D}, \forall t, \frac{dX_t}{dt} = (F_p + F_a)(X_t)$$

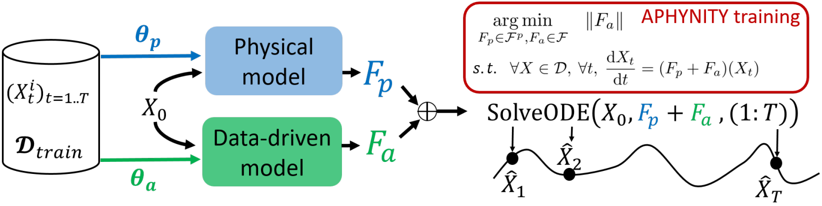


Theory: if $\mathcal{F}_p$ is a Chebyshev set[1], the decomposition exists and is unique

[1] in finite dim spaces, closed convex sets

APHYNITY - trajectory based training

- Parameterized models $F_p^{\theta_p}$ (θ_p physical parameters), $F_a^{\theta_a}$ (θ_a deep NN)



APHYNITY trajectory-based relaxation:

$$\min_{F_p \in \mathcal{F}_p, F_a \in \mathcal{F}} \ \|F_a\| + \lambda \ \ \mathbb{E}_{\mathcal{D}} \left[\int_t \|X_t - \tilde{X}_t\| dt \right] \quad \text{subject to}$$

$$\frac{d\tilde{X}_t}{dt} = (F_p + F_a)(\tilde{X}_t), \quad \tilde{X}_0 = X_0$$

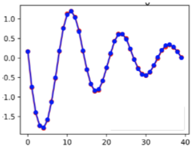
- Optimization with an increasing sequence of λ (variant of Uzawa algorithm)

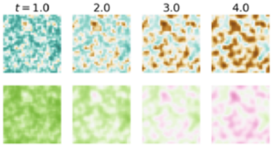
APHYNITY - experimental results

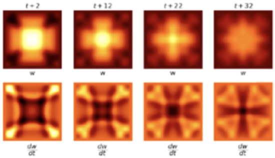
Experiments on 3 classes of physical phenomena:

- **Damped pendulum:** $\frac{d^2\theta}{dt^2} + \omega_0^2 \sin\theta + \lambda \frac{d\theta}{dt} = 0$
  - Simplified $\mathcal{F}_p$: Hamiltonian (energy conservation), ODE without λ
- **Reaction-diffusion:** $\frac{\partial u}{\partial t} = a\Delta u + R_u(u, v; k)$, $\frac{\partial v}{\partial t} = b\Delta v + R_v(u, v))$
  - Reaction terms: $R_u(u, v; k) = u - u^3 - k - v, R_v(u, v) = u - v$
  - Simplified $\mathcal{F}_p$: PDE without reaction
- **Damped wave**: $\frac{\partial^2 w}{\partial t^2} - c^2 \Delta w + k \frac{\partial w}{\partial t} = 0$
  - Simplified $\mathcal{F}_p$: PDE without damping

All $\mathcal{F}_p$ are closed and convex in $\mathcal{F}$ $\Rightarrow$ Chebyshev





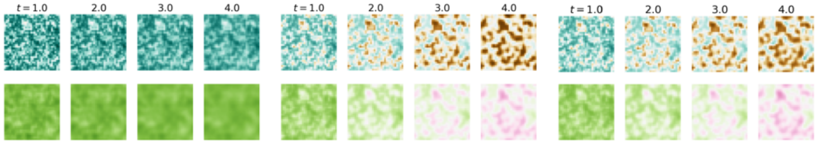


APHYNITY - quantitative results

APHYNITY successful complements simplified physical models for:

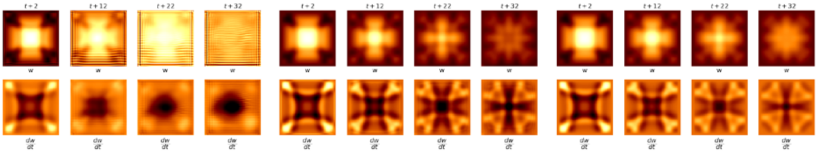
- better forecasting performances
- better physical parameter identification
- $\|F_a\|^2 \sim$ level of F_p physical approximation

	Method	$\log$ MSE	%Err param.	$\|F_a\|^2$
Incomplete physics	Hamiltonian	-0.35$\pm$0.10	n/a	n/a
	APHYNITY Hamiltonian	**-3.97$\pm$1.20**	n/a	623
	Param ODE (ω_0)	-0.14$\pm$0.10	13.2	n/a
	APHYNITY Param ODE (ω_0)	**-7.86$\pm$0.60**	**4.0**	132
Complete physics	True ODE	**-8.58$\pm$0.20**	n/a	n/a
	APHYNITY True ODE	**-8.44$\pm$0.20**	n/a	2.3



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

Figure: Predictions of u (top) and v for the reaction-diffusion equations.



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

Figure: Predictions on the damped wave equations.

Outline

Model-Based RL with prior physical models

- NeurIPS 2022 Offline RL workshop

Residual Model-Based Reinforcement Learning for Physical Dynamics

Zakariae EL ASRI
CEDRIC*
zakariae.el-asri@lecnam.net

Clément RAMBOUR
CEDRIC*
clement.rambour@lecnam.net

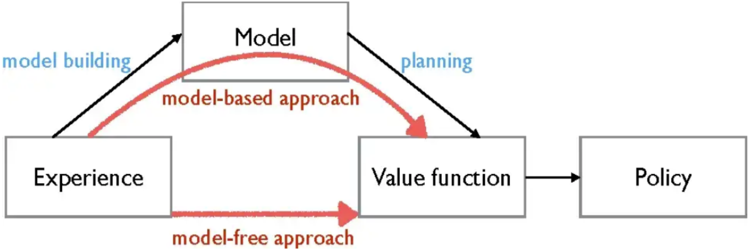
Nicolas THOME
ISIR[†]
nicolas.thome@sorbonne-universite.fr

Vincent Le Guen
EDF[‡]
vincent.le-guen@edf.fr

Abstract

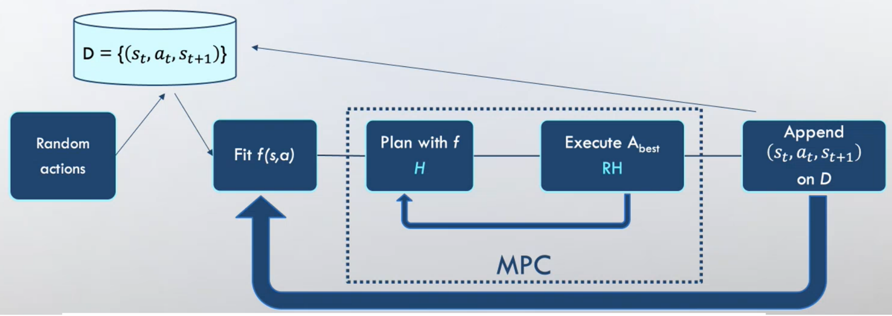
Dynamic control problems are a prevalent topic in robotics. Deep neural networks have been shown to learn accurately many complex dynamics, but these approaches remain data-inefficient or intractable in some tasks. Rather than learning to reproduce the environment dynamics, traditional control approaches use some physical knowledge to describe the environment's evolution. These approaches do not need

Model-Free *v.s.* Model-Based RL



Model-Free RL	*v.s.*	**Model-Based RL**
Directly learn a value function/policy with samples		Learn a transition model of the world, then plan with the model
+ very flexible / good performances - need a lot of samples		+ Sample efficient - accuracy of the model has a huge impact

Model-Based Reinforcement Learning (MBRL)



1 – Run random actions to collect dataset $D = \{(s_t, a_t, s_{t+1})\}$

2 – Learn dynamics model $f^{\theta}(s_t, a_t)$ to minimize loss function $\mathcal{L}(\theta)$

3 – Plan through the learned model $f^{\theta}(s_t, a_t)$ to choose actions over horizon H

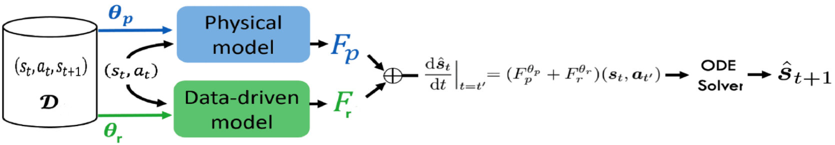
4 – Execute actions on a receding horizon RH $\ll$ H.

5 – Append (s_t, a_t, s_{t+1}) to dataset D

⚠ In MBRL, *model* often refers to data-driven models !

$\Rightarrow$ **In this work, physical models as prior**

Training the dynamical model



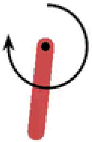
Training objective

$$\text{minimize} \quad \mathcal{L}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(s_t, a_t, s_{t+1}) \in \mathcal{D}} \|\hat{s}_{t+1} - s_{t+1}\|_2^2 + \alpha \|F_r^{\theta_r}\|_2$$

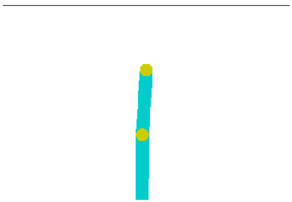
$$\text{subject to} \quad \frac{\mathrm{d}s_t}{\mathrm{d}t}\Big|_{t=t'} = (F_p^{\theta_p} + F_r^{\theta_r})(s_t, a_{t'}).$$

Experimental setup

- 2 standard control tasks from OpenAI Gym



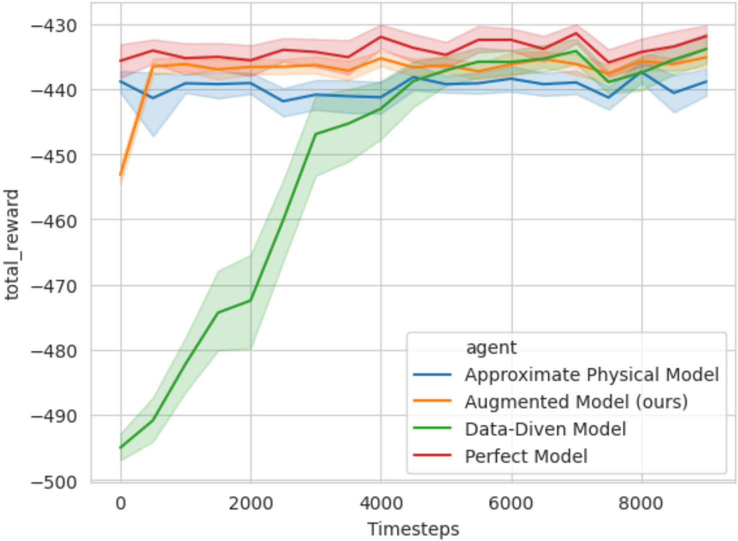
pendulum
$S \in \mathbb{R}^2, A \in \mathbb{R}$



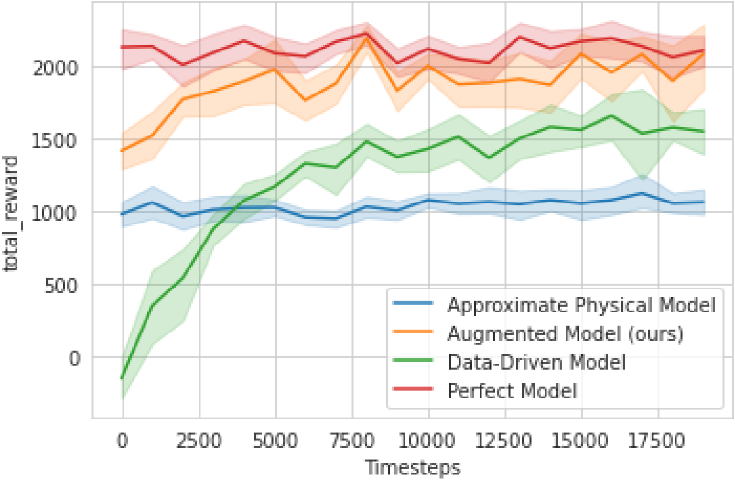
acrobot
$S \in \mathbb{R}^4, A \in \mathbb{R}^2$

- **Simplified physical models:** pendulum and acrobot equations without friction

Results - Pendulum

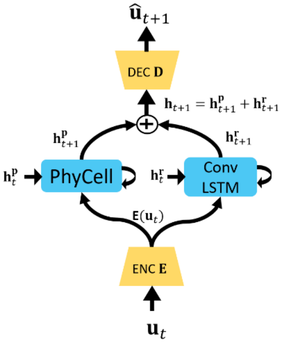


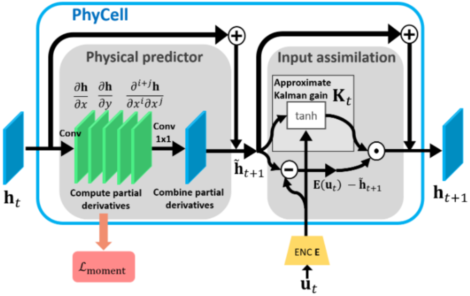
Results - Acrobot

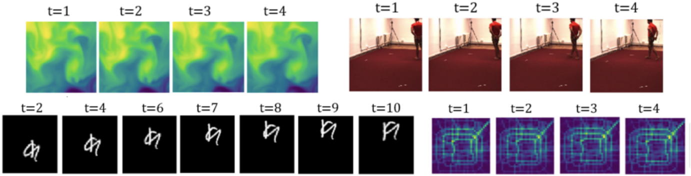


Outline

Application to video prediction: PhyDNet [CVPR'20]

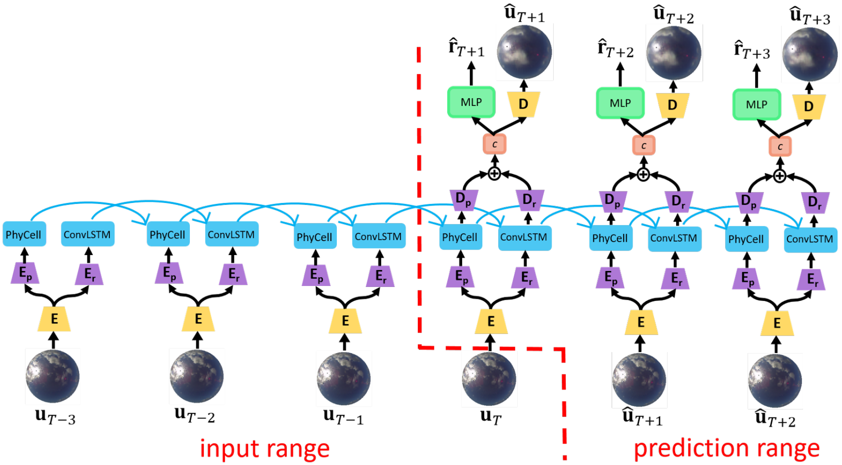






Photovoltaic energy forecasting [CVPR'20 workshop]

- Hybrid model for short term PV forecasting (0-20min) with fisheye images



Optical flow: traditional methods

- Traditional methods (eg. Horn Schunck, Lukas Kanade): based on the Brightness Constancy (BC) assumption

$$\frac{\partial I}{\partial t}(t, \mathbf{x}) + \mathbf{w}(t, \mathbf{x}) \cdot \nabla I(t, \mathbf{x}) = 0$$

- BUT: assumption fails in several common situations

Optical flow: deep supervised methods

- Deep models now state-of-the-art for optical flow, eg RAFT
- BUT need a complex training curriculum

Flying Chairs

Flying Things 3D

MPI Sintel

KITTI

Complementing the Brightness Constancy

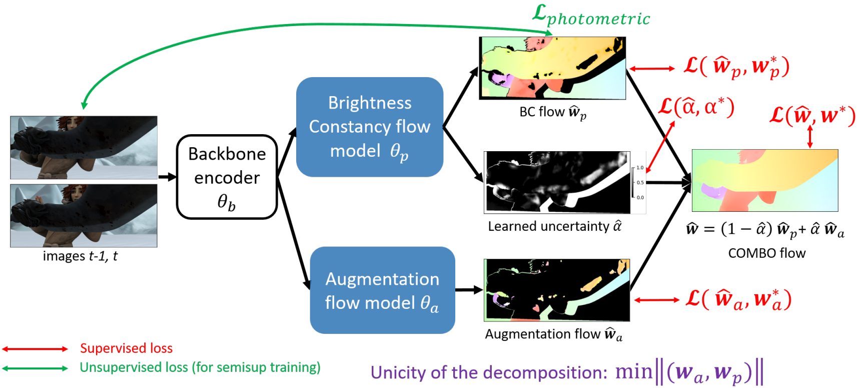
- Flow decomposition:

$$\underbrace{\mathbf{w}^*(\mathbf{x})}_{\text{GT optical flow}} = (1-\alpha^*(\mathbf{x})) \underbrace{\mathbf{w}_p^*(\mathbf{x})}_{\text{physical (BC) flow}} + \underbrace{\alpha^*(\mathbf{x})}_{\text{uncertainty of the BC}} \underbrace{\mathbf{w}_a^*(\mathbf{x})}_{\text{augmentation flow}}$$

- Constrained optimization problem to ensure a unique solution $(\mathbf{w}_p^*, \mathbf{w}_a^*, \alpha^*)$:

$$\min_{\mathbf{w}_p, \mathbf{w}_a} \quad \|(\mathbf{w}_a, \mathbf{w}_p)\| \qquad \text{subject to:}$$

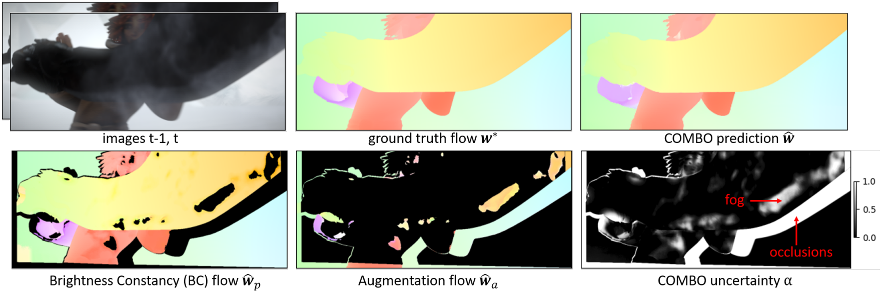
$$\begin{cases} (1-\alpha^*(\mathbf{x}))\ \mathbf{w}_p(\mathbf{x}) + \alpha(\mathbf{x})\ \mathbf{w}_a(\mathbf{x}) = \mathbf{w}^*(\mathbf{x}) \\ (1-\alpha^*(\mathbf{x}))\ |I_1(\mathbf{x}) - I_2(\mathbf{x}+\mathbf{w}_p(\mathbf{x}))| = 0 \\ \alpha^*(\mathbf{x}) = \sigma\left(|I_1(\mathbf{x}) - I_2(\mathbf{x}+\mathbf{w}^*(\mathbf{x}))|\right). \end{cases}$$



Semi-supervised: much simpler training curriculum

COMBO experimental results

- State-of-the-art results compared to RAFT in supervised training
- In the semi-supervised setting, greatly reduces the training curriculum to achieve similar performances



Outline

Conclusions and perspectives

- **Exploiting approximate physical knowledge in ML models**
  - Decomposition strategy with uniqueness guarantee (APHYNITY)
- **Benefits of physical priors in Model-Based RL**
  - Better sample efficiency while maintaining performances
  - Better model $\Rightarrow$ need to replan less frequently

Perspectives:

- Many interesting applications for physics-inspired / augmented models
  - fluid dynamics, electromagnetism, thermodynamics, ...
  - a part is explored in SINCLAIR lab
  - APHYNITY: what happens for non differentiable physical models ?
- Model-Based RL: application to real-world use cases

Conclusions and perspectives

- **Exploiting approximate physical knowledge in ML models**
  - Decomposition strategy with uniqueness guarantee (APHYNITY)
- **Benefits of physical priors in Model-Based RL**
  - Better sample efficiency while maintaining performances
  - Better model $\Rightarrow$ need to replan less frequently

Perspectives:

- Many interesting applications for physics-inspired / augmented models
  - fluid dynamics, electromagnetism, thermodynamics, ...
  - a part is explored in SINCLAIR lab
  - APHYNITY: what happens for non differentiable physical models ?
- Model-Based RL: application to real-world use cases

References I

[Chen et al., 2018] Chen, T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018).
Neural ordinary differential equations.
In *Advances in neural information processing systems (NeurIPS)*, pages 6571–6583.

[Corpetti et al., 2009] Corpetti, T., Héas, P., Mémin, E., and Papadakis, N. (2009).
Pressure image assimilation for atmospheric motion estimation.
Tellus A: Dynamic Meteorology and Oceanography, 61(1):160–178.

[de Bezenac et al., 2018] de Bezenac, E., Pajot, A., and Gallinari, P. (2018).
Deep learning for physical processes: Incorporating prior scientific knowledge.
International Conference on Learning Representations (ICLR).

[Raissi, 2018] Raissi, M. (2018).
Deep hidden physics models: Deep learning of nonlinear partial differential equations.
The Journal of Machine Learning Research, 19(1):932–955.

[Rico-Martinez et al., 1994] Rico-Martinez, R., Anderson, J., and Kevrekidis, I. (1994).
Continuous-time nonlinear signal processing: a neural network based approach for gray box identification.
In *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, pages 596–605. IEEE.

[Shi et al., 2015] Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015).
Convolutional lstm network: A machine learning approach for precipitation nowcasting.
Advances in neural information processing systems, 28.