Hybrid Dynamical Systems
Augmenting Physics with Machine Learning

Nicolas Thome - Prof. at Cnam Paris
CEDRIC Lab, Machine Learning Team (Vertigo)

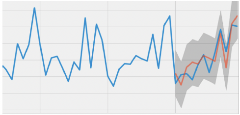
AI Chair OceaniX

April 20, 2022

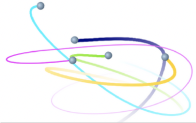
Outline

Spatio-temporal forecasting

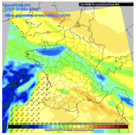
- **Future prediction** of time series with potential spatial correlations
- **Various tasks and applications:** weather and climate science, finance, healthcare, physics, robotics, *etc*
  - Climate: anticipating floods, hurricanes, earthquakes or other extreme events



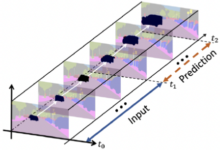
time series forecasting



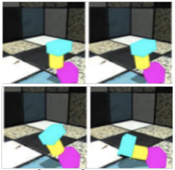
particle physics



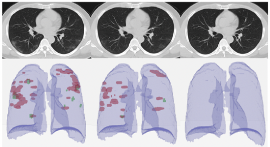
weather forecasts



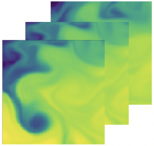
video prediction



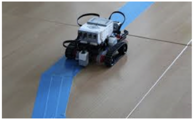
physical reasoning



medical prognosis , *e.g.* COVID evolution

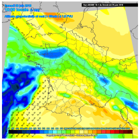


sea surface temperature

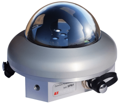


robot visual navigation

- **Big data:** superabundance of data: times series (sensor measurements), images (fisheye, satellite), spatio-temporal data (weather forecasts), *etc*

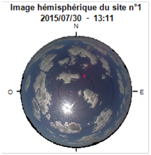


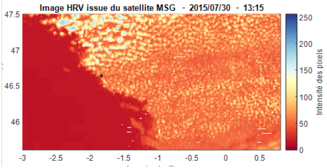
weather forecasts

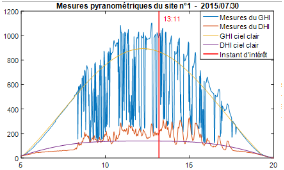


Sensors, pyranometers

100M monitoring cameras



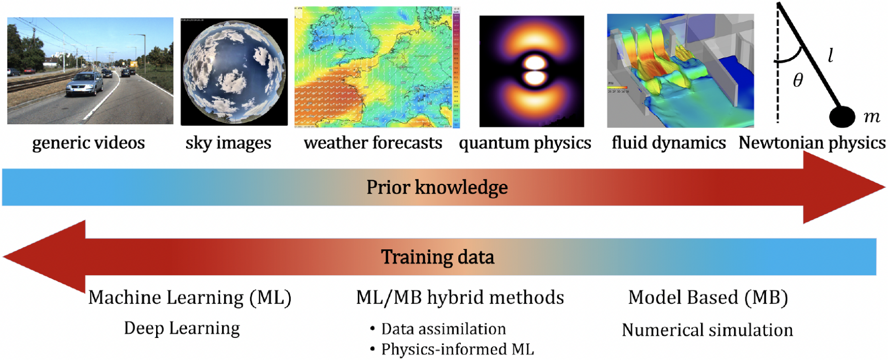




- Obvious need for **Artificial Intelligence** with these data

State-of-the-art in complex dynamic forecasting

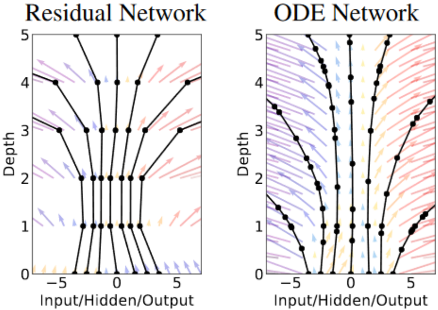
- **Model-Based (MB) approaches**, *e.g.* using PDE/ODE: deep understanding of complex underlying phenomenon
- **Machine Learning (ML) / Deep learning (DL)**: more agnostic, now state-of-the art in several tasks, *e.g.* ConvLSTM [9], Neural ODE [2]
  - **Hybrid MB/ML models: hot topic in the current deep learning era**



Neural ODE [2]

- Parameterizing ODE state derivative $\frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta)$ using a NN instead of a discrete sequence of hidden layers

Residual network	ODE network
$\mathbf{h}_{t+1} = \mathbf{h}_t + f(\mathbf{h}_t, \theta_t)$	$\frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta)$

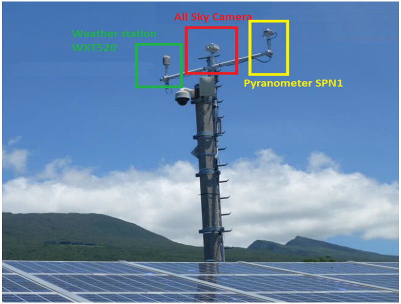


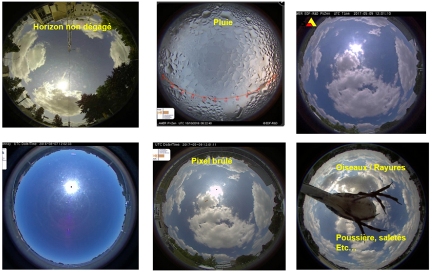
- Backward pass with the adjoint method
- **<u>BUT:</u> dynamic still purely data-driven, no physical knowledge**

Solar energy forecasting

Industrial application at EDF: short-term solar energy forecasting

Data: > 7 Million Fisheye images and measured solar irradiance every 10s





Goals:

1. Predict solar irradiance from fisheye image: **perception, works well**
2. Predict future irradiance (0-20min) given past images: **more challenging**

Challenges in solar energy forecasting

- Data-driven forecasts struggle to properly extrapolate
  - Lag behind GT, unable to capture sharp changes

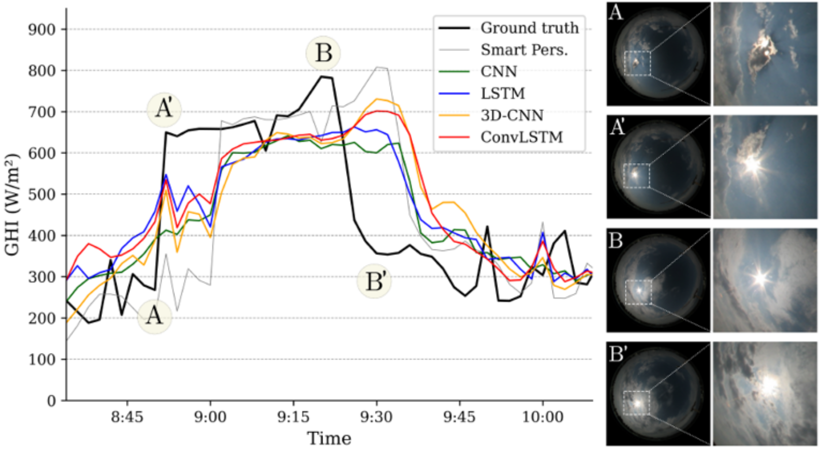
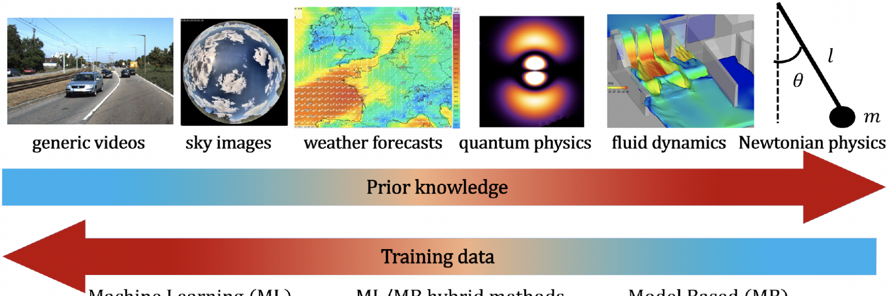


Figure: 5min solar irradiance forescasting (from Paletta et al. [7])

Research directions to improve forecasts



Machine Learning (ML)

Deep Learning

ML/MB hybrid methods

- Data assimilation
- Physics-informed ML

Model Based (MB)

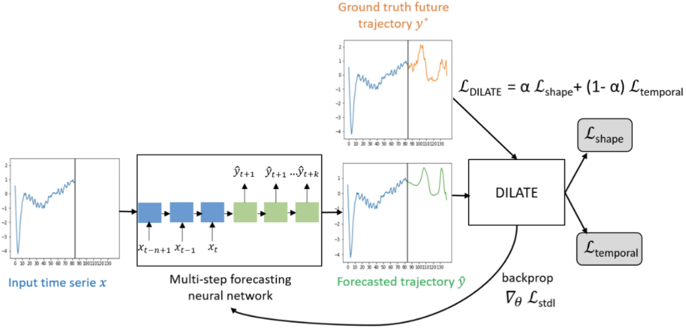
Numerical simulation

How to properly exploit prior physical knowledge to improve Machine Learning forecasting models?

Combine MB and ML models, hybrid models (gray box)
$\Rightarrow$ **Physically-constrained deep forecasting**

- Loss function regularization, *e.g.* Physics-Informed Neural Networks (PINNs) [8] - Using shape and time criterion, V. Le Guen's PhD [4, 5]

$$\mathcal{L}_{DILATE}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) = \alpha\ \mathcal{L}_{shape}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) + (1-\alpha)\ \mathcal{L}_{temporal}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i)$$



Incorporating prior information in machine learning

Combine MB and ML models, hybrid models (gray box)
$\Rightarrow$ **Physically-constrained deep forecasting**

- Loss function regularization, *e.g.* Physics-Informed Neural Networks (PINNs) [8]
- Contraints in deep architecture [3, 6]

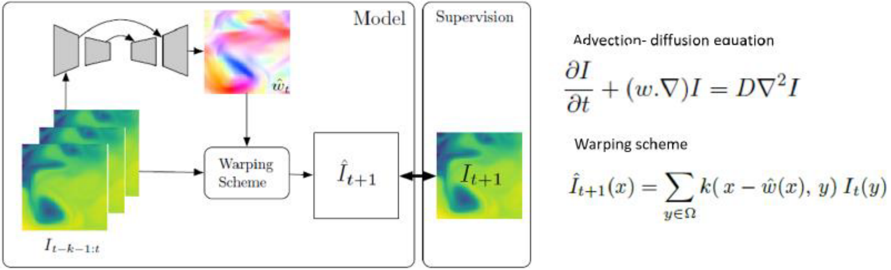


Figure: Advection-diffusion flow [3]

PDE-Net [6]

NN architecture to approximate the solution of a linear PDE

$$\frac{\partial u}{\partial t}(t,x,y) = F(x,y,\frac{\partial u}{\partial x},\frac{\partial u}{\partial y},\frac{\partial^2 u}{\partial x^2},\dots) = \sum_{(i,j):i+j\leq q} c_{i,j}\frac{\partial^{i+j}\mathbf{h}}{\partial x^i \partial y^j}(t,\mathbf{x}) \quad (1)$$

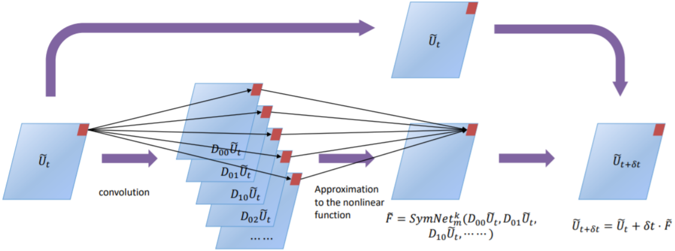


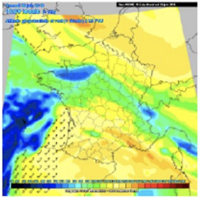
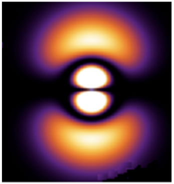
Figure 1: The schematic diagram of a δt-block.

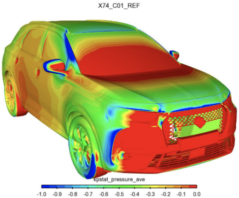
- Euler discretization: $u(t+1) \approx u(t) + \delta t.F(x,y,u_x(t),\dots) \Rightarrow$ approximate derivatives with a residual architecture and constrained convolutions
- Estimate unknown ODE parameters, *e.g.* $\{c_{i,j}\}$ in Eq (1)

$\Rightarrow$ **Physical parameters identification**

Focus & Contributions

- **Physical models: approximations of real world dynamics**

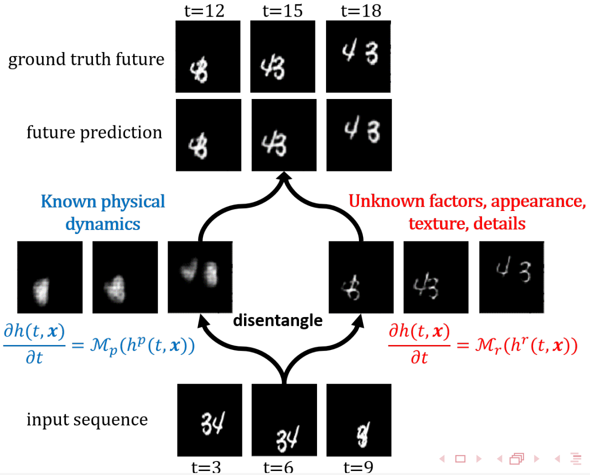


$\Rightarrow$ **Augmenting simplified physical models with data-driven networks**

- Fully vs partial observability: physical models not always applicable in input space
  $\Rightarrow$ **Leveraging PDE dynamics in a learned latent space**
- Proper cooperation between prior physical and data-driven augmentation
  $\Rightarrow$ **Decomposition with uniqueness guarantees**

Outline

Disentangling Physics, *e.g.* PDE, from complementary info in latent space



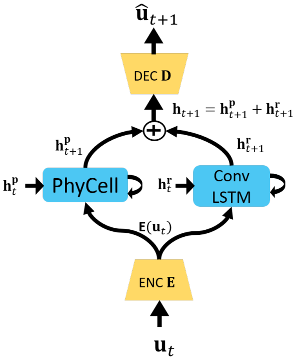
PhyDNet

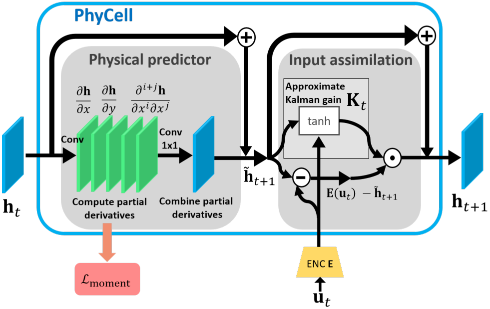
Modeling physics in latent space

- input image: $\mathbf{u}(t, \mathbf{x})$
- $\mathbf{E}[\mathbf{u}(t, \mathbf{x})] = \mathbf{h}(t, \mathbf{x})$ in latent space $\mathcal{H}$
- **PDE dynamics in $\mathcal{H}$:**

$$\frac{\partial \mathbf{h}(t, \mathbf{x})}{\partial t} = \frac{\partial \mathbf{h}^{\mathbf{p}}}{\partial t} + \frac{\partial \mathbf{h}^{\mathbf{r}}}{\partial t} := \mathcal{M}_p(\mathbf{h}^{\mathbf{p}}, \mathbf{u}) + \mathcal{M}_r(\mathbf{h}^{\mathbf{r}}, \mathbf{u})$$

- **Latent dynamics decomposed into:**
  - $\frac{\partial \mathbf{h}^{\mathbf{p}}}{\partial t}$ **with physical prior $\Rightarrow$ PhyCell**
  - **Data-driven augmentation: ConvLSTM**
- Such that: $\mathbf{D}[\mathbf{h}(t+1, \mathbf{x}))] \approx \mathbf{u}(t+1, \mathbf{x})$





Prediction-correction revisited with deep learning

- Decoupling prediction/correction: good for long-term forecasting and missing data

Prediction: $\tilde{\mathbf{h}}_{t+1} = \mathbf{h}_t + \Phi(\mathbf{h}_t)$

- $\Phi(\mathbf{h}(t,\mathbf{x})) = \sum_{i,j:i+j \leq q} c_{i,j} \dfrac{\partial^{i+j}\mathbf{h}}{\partial x^i \partial y^j}(t,\mathbf{x})$, PDE-Net in latent space
- Approximate $\dfrac{\partial^{i+j}\mathbf{h}}{\partial x^i \partial y^j}$ with conv & moment loss
- $\{c_{i,j}\}$ learned

Correction:

$$\mathbf{h}_{t+1} = \tilde{\mathbf{h}}_{t+1} + \mathbf{K}_t \odot \left(\mathbf{E}(\mathbf{u}_t) - \tilde{\mathbf{h}}_{t+1}\right)$$

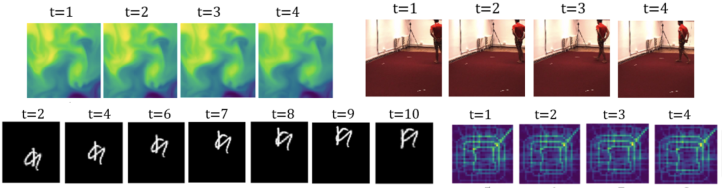
- Learned Kalman gain $\mathbf{K}_t$ for correction: $\mathbf{K}_t = \tanh\left(\mathbf{W}_h * \tilde{\mathbf{h}}_{t+1} + \mathbf{W}_u * \mathbf{E}(\mathbf{u}_t) + \mathbf{b}_k\right)$
- $\neq$ locally linear approximate Kalman gain in [1]

State-of-the-art wrt

- Recent general video prediction methods
- specialized models for Moving Mnist and SST

	Moving MNIST			Traffic BJ			Sea Surface Temperature			Human 3.6		
Method	MSE	MAE	SSIM	MSE ×100	MAE	SSIM	MSE ×10	MAE	SSIM	MSE / 10	MAE /100	SSIM
ConvLSTM [73]	103.3	182.9	0.707	48.5*	17.7*	0.978*	45.6*	63.1*	0.949*	50.4*	18.9*	0.776*
PredRNN [66]	56.8	126.1	0.867	46.4	17.1*	0.971*	41.9	62.1	0.955	48.4	18.9	0.781
Causal LSTM [64]	46.5	106.8	0.898	44.8	16.9*	0.977*	39.1*	62.3*	0.929*	45.8	17.2	0.851
MIM [67]	44.2	101.1	0.910	42.9	16.6*	0.971*	42.1*	60.8*	0.955*	42.9	17.8	0.790
E3D-LSTM [65]	41.3	86.4	0.920	43.2*	16.9*	0.979*	34.7*	59.1*	0.969*	46.4	16.6	0.869
Advection-diffusion [11]	-	-	-	-	-	-	34.1*	54.1*	0.966*	-	-	-
DDPAE [21]	38.9	90.7*	0.922*	-	-	-	-	-	-	-	-	-
PhyDNet	**24.4**	**70.3**	**0.947**	**41.9**	**16.2**	**0.982**	**31.9**	**53.3**	**0.972**	**36.9**	**16.2**	**0.901**

Table 1. Quantitative forecasting results of PhyDNet compared to baselines using various datasets. Numbers are copied from original or citing papers. * corresponds to results obtained by running online code from the authors. The first five baseline are general deep models applicable to all datasets, whereas DDPAE [21] (resp. advection-diffusion flow [11]) are specific state-of-the-art models for Moving MNIST (resp. SST). Metrics are scaled to be in a similar range across datasets to ease comparison.



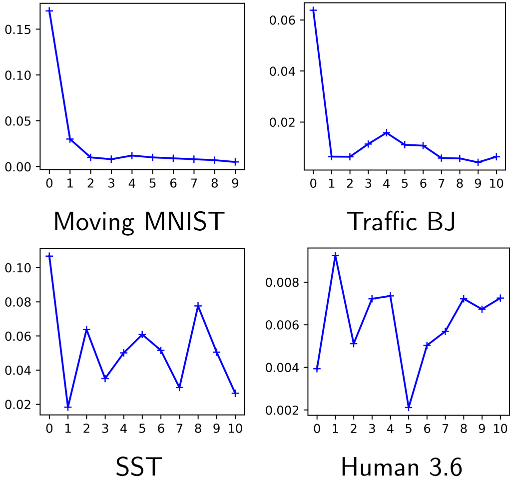
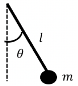


Figure: Mean amplitude of the combining coefficients $c_{i,j}$ with respect to the order of the differential operators approximated.

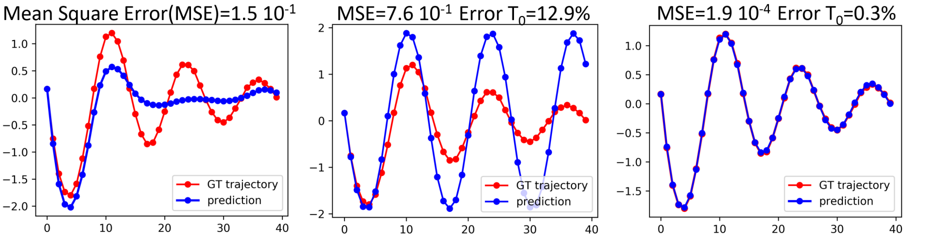
Outline

Motivation: data-driven vs. simplified physical models

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



- **Data-driven models** struggle to extrapolate complex dynamics, in particular in data-scarce contexts
- **Physical models** fail to extrapolate when they are misspecified: forecasting & parameter identification failure



(a) Data-driven Neural ODE (b) Simple physical model (c) Our APHYNITY framework

$\Rightarrow$ **Augmenting PHYsical models for ideNtIfying and forecasTing complex dYnamic (APHYNITY)**

APHYNITY

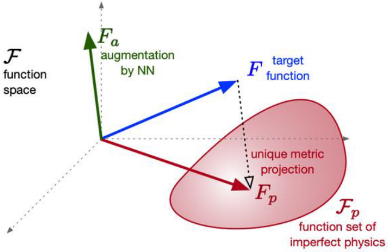
- $\frac{\mathrm{d}X_t}{\mathrm{d}t} = F(X_t)$, $X_t \in \mathbb{R}^d$ (vector) or $X_t(\mathbf{x}) \in \mathbb{R}^d$, $\mathbf{x} \in \Omega \subset \mathbb{R}^k$ (vector field)
  - $F \in \mathcal{F}$ normed vector space, $F_p \in \mathcal{F}_p \subset \mathcal{F}$ physical model (ODE/PDE)
- **Augment approximate physical model F_p with data-driven $F_a \in \mathcal{F}$:**

$$\boxed{\frac{dX_t}{dt} = F(X_t) = F_p + F_a}$$

- However, decomposition $F = F_p + F_a$ in general not unique
- APHYNITY:

$$\min_{F_p \in \mathcal{F}_p, F_a \in \mathcal{F}} \|F_a\| \text{ subject to } F = (F_p + F_a) \quad (2)$$

  - If $\mathcal{F}_p$ Chebyshev set[a], decomposition in Eq (2) exists and is unique (metric projection onto $\mathcal{F}_p$).



Intuition: min $||F_a|| \Rightarrow$ augmentation only models information that cannot be captured by the physical prior F_p

[a] In finite-dim space, closed convex sets

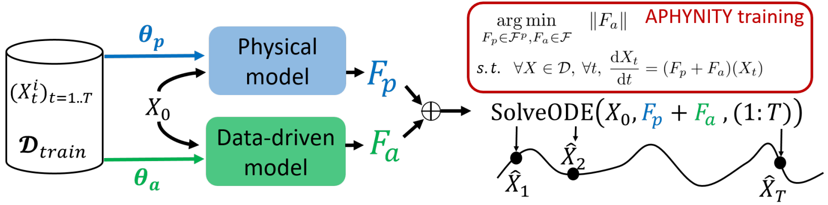
APHYNITY training

- Dataset of observed trajectories: $\mathcal{D} = \{X_\cdot : [0,T] \to \mathcal{F} \mid \forall t \in [0,T], {}^{\mathrm{d}X_t}/_{\mathrm{d}t} = F(X_t)\}$

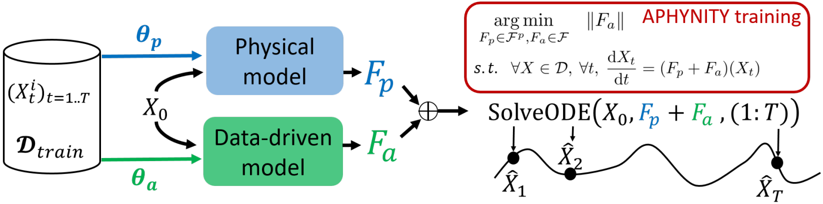
APHYNITY objective:

$$\min_{F_p \in \mathcal{F}_p, F_a \in \mathcal{F}} \|F_a\| \quad \text{subject to} \quad \forall X \in \mathcal{D}, \forall t, \frac{dX_t}{dt} = (F_p + F_a)(X_t)$$

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



APHYNITY optimization



- **Trajectory based training:** multi-step prediction, differentiable ODE solver
- In practice: adaptive constraint optimization (variant of Uzawa algorithm):

$$\mathcal{L}_{\lambda_j}(\theta_p, \theta_a) = \|F_a^{\theta_a}\| + \lambda_j \cdot \mathcal{L}_{traj}(\theta_p, \theta_a) \tag{3}$$

  - $\mathcal{L}_{traj}(\theta_p, \theta_a) = \sum_{i=1}^{N} \sum_{h=1}^{T/\Delta t} \|X_{h\Delta t}^{(i)} - \widetilde{X}_{h\Delta t}^{(i)}\|$

- $\theta = (\theta_p, \theta_a)$, Iterative λ_j setting:
  - $\lambda_{j+1} = \lambda_j + \tau_2 \mathcal{L}_{traj}(\theta_{j+1})$, τ_2 hyper-parameter
- Stable and robust convergence

Algorithm 1: APHYNITY

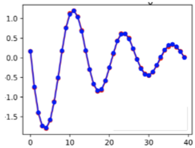
Initialization: $\lambda_0 \geq 0, \tau_1 > 0, \tau_2 > 0$;
for *epoch* $= 1 : N_{epochs}$ **do**
  for *iter in* $1 : N_{iter}$ **do**
    for *batch in* $1 : B$ **do**
      $\theta_{j+1} = \theta_j - \tau_1 \nabla [\lambda_j \mathcal{L}_{traj}(\theta_j) + \|F_a\|]$
  $\lambda_{j+1} = \lambda_j + \tau_2 \mathcal{L}_{traj}(\theta_{j+1})$

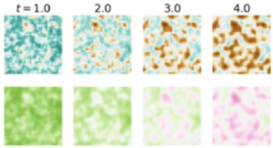
APHYNITY - quantitative results

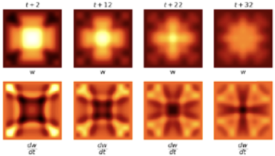
Experiments on 3 classes of physical phenomena:

- **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 $\mathcal{F}_p$'s are closed and convex in $\mathcal{F}$ $\Rightarrow$ Chebyshev





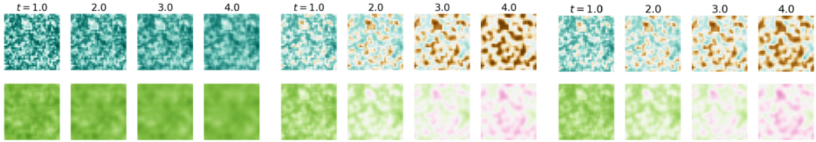


Experiments: APHYNITY results

Dataset		Method	log MSE	%Err param.	$\|\|F_a\|\|^2$
(a) Reaction-diffusion	Data-driven	Neural ODE	-3.76±0.02	n/a	n/a
		PredRNN++	-4.60±0.01	n/a	n/a
	Incomplete physics	Param PDE (a, b)	-1.26±0.02	67.6	n/a
		APHYNITY Param PDE (a, b)	**-5.10±0.21**	**2.3**	67
	Complete physics	Param PDE (a, b, k)	**-9.34±0.20**	0.17	n/a
		APHYNITY Param PDE (a, b, k)	**-9.35±0.02**	**0.096**	1.5e-6
		True PDE	-8.81±0.05	n/a	n/a
		APHYNITY True PDE	**-9.17±0.02**	n/a	1.4e-7
(b) Wave equation	Data-driven	Neural ODE	-2.51±0.29	n/a	n/a
	Incomplete physics	Param PDE (c)	0.51±0.07	10.4	n/a
		APHYNITY Param PDE (c)	**-4.64±0.25**	**0.31**	71.
	Complete physics	Param PDE (c, k)	-4.68±0.55	1.38	n/a
		APHYNITY Param PDE (c, k)	**-6.09±0.28**	**0.70**	4.54
		True PDE	-4.66±0.30	n/a	n/a
		APHYNITY True PDE	**-5.24±0.45**	n/a	0.14
(c) Damped pendulum	Data-driven	Neural ODE	-2.84±0.70	n/a	n/a
	Incomplete physics	Hamiltonian	-0.35±0.10	n/a	n/a
		APHYNITY Hamiltonian	**-3.97±1.20**	n/a	623
		Param ODE (ω_0)	-0.14±0.10	13.2	n/a
		Deep Galerkin Method (ω_0)	-3.10±0.40	22.1	n/a
		APHYNITY Param ODE (ω_0)	**-7.86±0.60**	**4.0**	132
	Complete physics	Param ODE (ω_0, α)	**-8.28±0.40**	**0.45**	n/a
		Deep Galerkin Method (ω_0, α)	-3.14±0.40	7.1	n/a
		APHYNITY Param ODE (ω_0, α)	**-8.31±0.30**	**0.39**	8.5
		True ODE	**-8.58±0.20**	n/a	n/a
		APHYNITY True ODE	**-8.44±0.20**	n/a	2.3

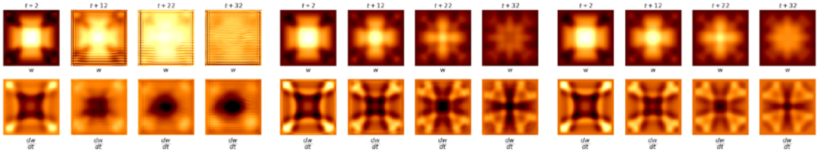
- Better forecasting performances
- Better physical parameter identification
- $\|\|F_a\|\|^2$ ~ level of F_p approximation

APHYNITY - qualitative results



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

Figure 3: Comparison of predictions of two components u (top) and v (bottom) of the reaction-diffusion system. Note that $t = 4$ is largely beyond the dataset horizon $(t = 2.5)$.



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

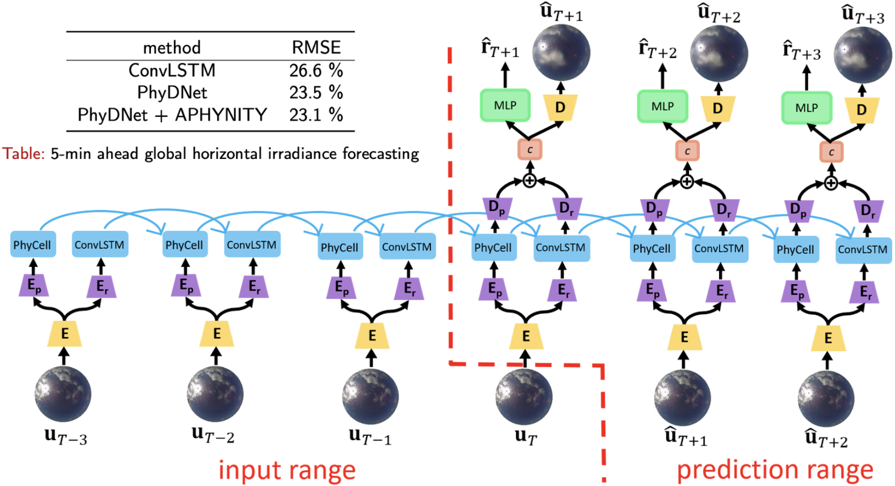
Figure 4: Comparison between the prediction of APHYNITY when c is estimated and Neural ODE for the damped wave equation. Note that $t + 32$ is already beyond the dataset horizon $(t + 25)$, showing the consistency of APHYNITY method.

Application to solar energy forecasting (CVPR'20 workshop)

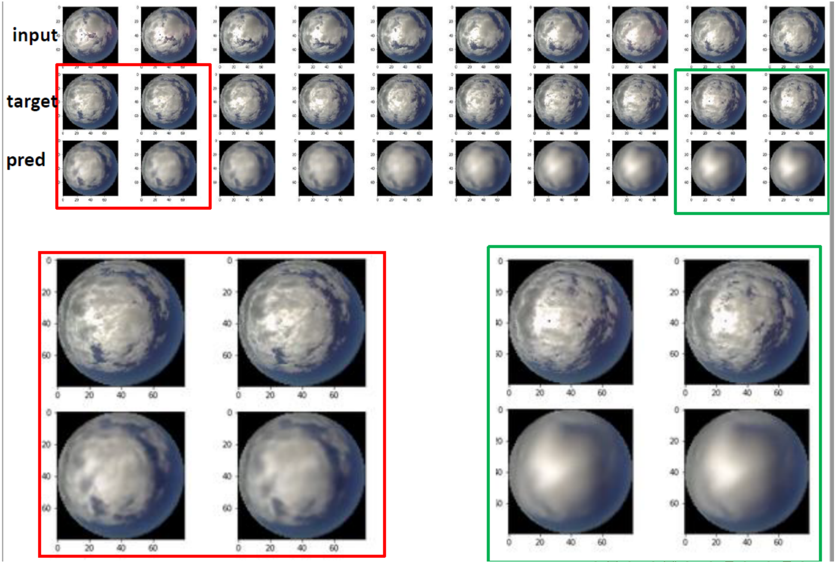
- Short-term (<20min) solar irradiance forecasting with fisheye images
- Improved PhyDNet model with separate encoders/decoders & min $||F_a||^2$

method	RMSE
ConvLSTM	26.6 %
PhyDNet	23.5 %
PhyDNet + APHYNITY	23.1 %

Table: 5-min ahead global horizontal irradiance forecasting



Qualitative results



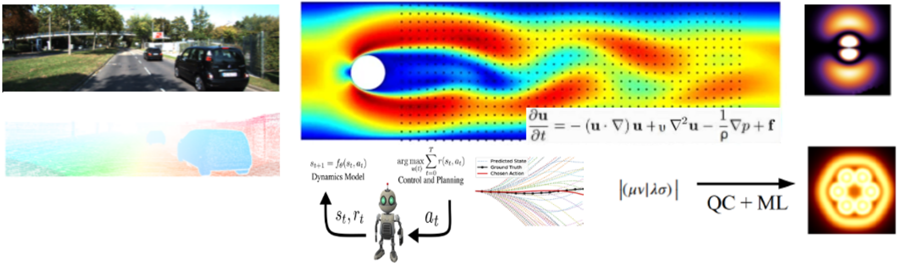
Conclusions and perspectives

New hybrid MB/ML models

- **Leveraging approximate physics to improve deep forecasting models**
- Importance of proper physics/ML decomposition, improving forecasting performances and parameter identification

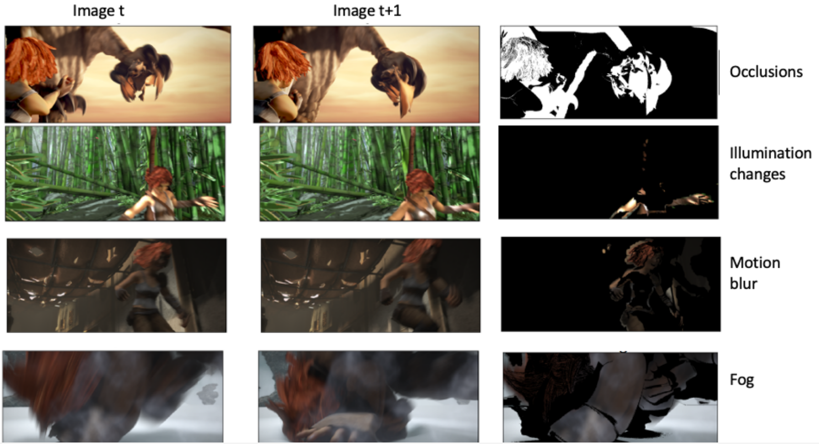
Future works:

- Application in other applications domains: optical flow, computational fluid dynamics, control (model-based RL), quantum physics
- Beyond summing derivatives: $F = F_p + F_a$



Physical models for optical flow

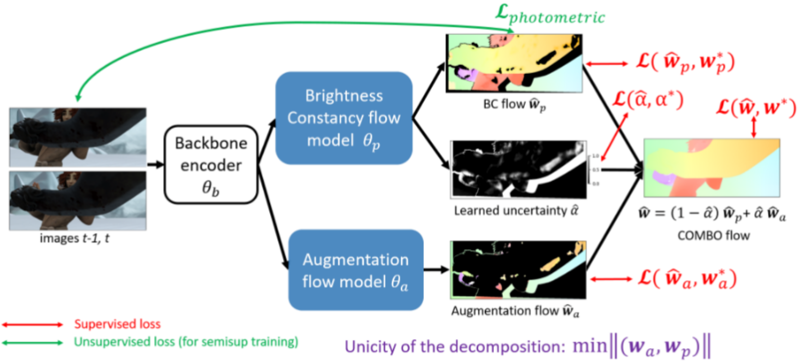
- Deep learning models: trained with complex curriculum, *i.e.* synthetic data (Chairs, Things, Sintel), real data (HD1K, Kitti)
- Traditional methods: based on brightness consistency (BC) assumption: $\frac{\partial I}{\partial t}(t,\mathbf{x}) + \mathbf{w}(t,\mathbf{x}) \cdot \nabla I(t,\mathbf{x}) = 0$
  - BUT: BC violated in several usual conditions



COMBO model for optical flow

- COMBO: complementing BC with deep NNs for accurate flow prediction
- GT flow $\mathbf{w}^*$ decomposition: physical flow $\mathbf{w}_p^*$, augmentation flow $\mathbf{w}_a^*$, uncertainty map α^*: $\min_{\mathbf{w}_p,\mathbf{w}_a} \|(\mathbf{w}_a, \mathbf{w}_p)\|$ subject to: (4)

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



Unicity of the decomposition: $\min\|(w_a, w_p)\|$

Thank your for your attention!

Questions?

Augmented physical models:

- **PhyDNet:** V. Le Guen, N. Thome
  - **CVPR'20 paper**, **CVPR'20 OMNI-CV workshop**
  - **GitHub code:** `https://github.com/vincent-leguen/PhyDNet`
- **APHYNITY:** Y. Yin, V. Le Guen, J. Dona, I. Ayed, E. de Bézenac, N. Thome, P. Gallinari
  - **ICLR'21 paper**, **GitHub code:** `https://github.com/yuan-yin/aphynity`
- **COMBO:** V. Le Guen, C. Rambour, N. Thome. ECCV'22 submission

Loss function regularization: V. Le Guen, N. Thome

- **T-PAMI'22 paper**: deep time series forecasting with shape & temporal criteria
- **DILATE: NeurIPS'19**, **GitHub:** `https://github.com/vincent-leguen/DILATE`
- **STRIPE: NeurIPS'20**, **GitHub code:** `https://github.com/vincent-leguen/STRIPE`

References I

[1] Philipp Becker, Harit Pandya, Gregor Gebhardt, Cheng Zhao, C James Taylor, and Gerhard Neumann.
Recurrent Kalman networks: Factorized inference in high-dimensional deep feature spaces.
In *International Conference on Machine Learning (ICML)*, pages 544–552, 2019.

[2] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud.
Neural ordinary differential equations.
In *Advances in neural information processing systems (NeurIPS)*, 2018.

[3] Emmanuel de Bezenac, Arthur Pajot, and Patrick Gallinari.
Deep learning for physical processes: Incorporating prior scientific knowledge.
International Conference on Learning Representations (ICLR), 2018.

[4] Vincent Le Guen and Nicolas Thome.
Shape and time distortion loss for training deep time series forecasting models.
In *Advances in Neural Information Processing Systems*, pages 4191–4203. 2019.

[5] Vincent Le Guen and Nicolas Thome.
Probabilistic Time Series Forecasting with Structured Shape and Temporal Diversity.
In *NeurIPS 2020*, Vancouver, Canada, December 2020.

[6] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong.
PDE-Net: Learning PDEs from data.
In *International Conference on Machine Learning*, pages 3214–3222, 2018.

[7] Quentin Paletta, Guillaume Arbod, and Joan Lasenby.
Benchmarking of deep learning irradiance forecasting models from sky images–an in-depth analysis.
arXiv preprint arXiv:2102.00721, 2021.

[8] M. Raissi, P. Perdikaris, and G. E. Karniadakis.
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.

References II

[9] Shi Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems (NeurIPS)*, pages 802–810, 2015.