

Augmenting Incomplete Physical Models with Deep Networks for Complex Dynamic Forecasting

**Nicolas Thome - Prof. at Cnam Paris
CEDRIC Lab, Vertigo Team**

**Fall Seminar, Department of Energy Resources Engineering,
Stanford University**

October 4, 2021

Stanford
EARTH

SCHOOL OF EARTH, ENERGY
& ENVIRONMENTAL SCIENCES

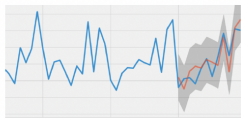


Outline

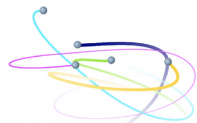
- 1 Context
- 2 PhyDNet: Physically-constrained deep video prediction
- 3 APHYNITY: physics & ML cooperation

Spatio-temporal forecasting

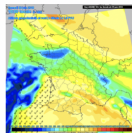
- ▶ **Future prediction** of time series with potential spatial correlations
- ▶ **Various tasks and applications:** weather and climate science, finance, healthcare, physics, robotics, *etc*
 - ▶ Climate: anticipating floods, hurricanes, earthquakes or other extreme events



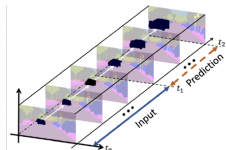
time series forecasting



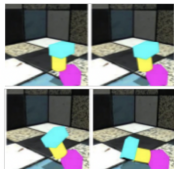
particle physics



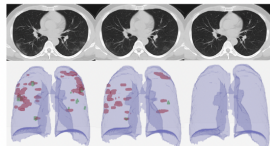
weather forecasts



video prediction



physical reasoning



medical prognosis, *e.g.* COVID evolution



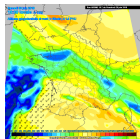
sea surface temperature



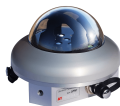
robot visual navigation

Spatio-temporal forecasting & big data

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



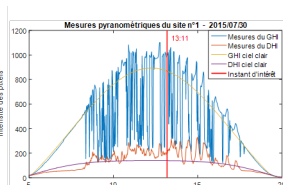
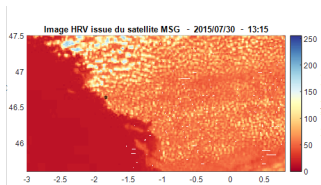
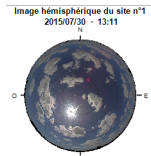
weather forecasts



Sensors, pyranometers



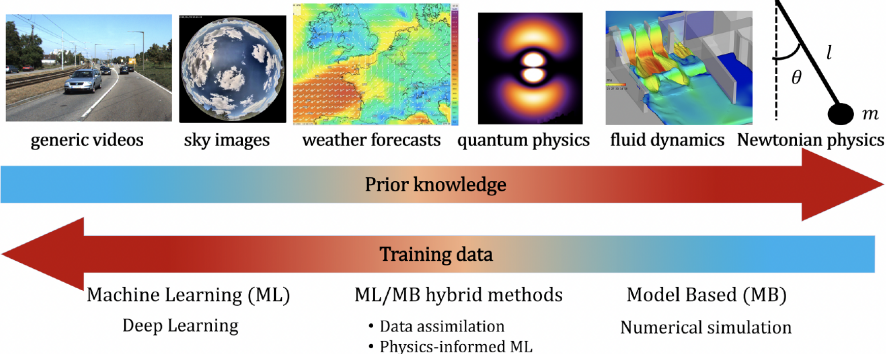
100M monitoring cameras



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

State-of-the-art in complex dynamic forecasting

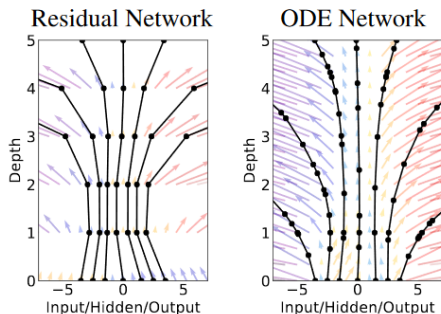
- ▶ **Model-Based (MB) approaches**, e.g. using PDE/ODE: deep understanding of complex underlying phenomenon
- ▶ **Machine Learning (ML) / Deep learning (DL)**: more agnostic, now state-of-the-art in several tasks, e.g. ConvLSTM [17], Neural ODE [2]
 - ▶ Hybrid MB/ML models: hot topic in the current 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

$$\begin{array}{l|l} \text{Residual network} & \text{ODE network} \\ \hline \mathbf{h}_{t+1} = \mathbf{h}_t + f(\mathbf{h}_t, \theta_t) & \frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta) \end{array}$$

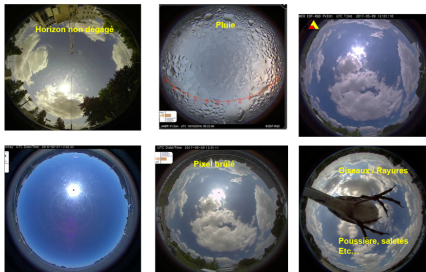
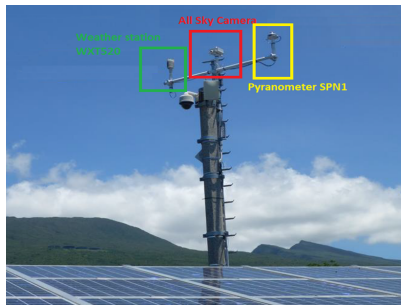


- ▶ Backward pass with the adjoint method
- ▶ **BUT**: dynamic still purely data-driven, no physical knowledge

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

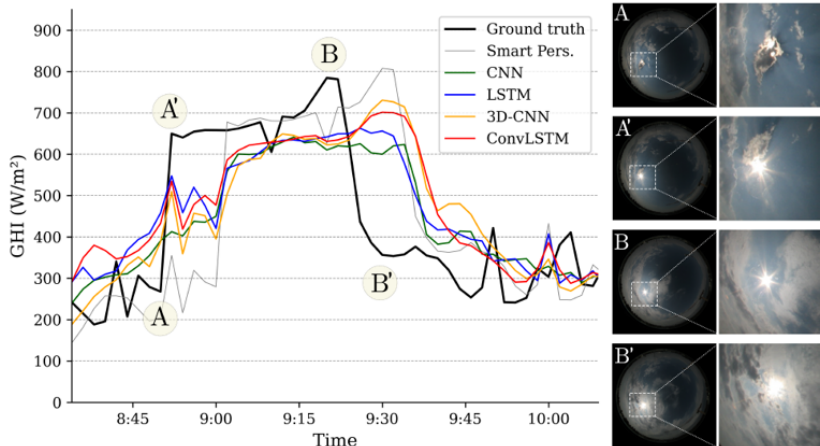
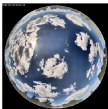


Figure: 5min solar irradiance forecasting (from Paletta et al. [8])

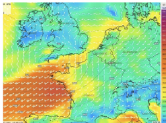
Research directions to improve forecasts



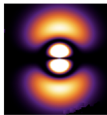
generic videos



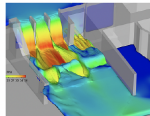
sky images



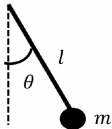
weather forecasts



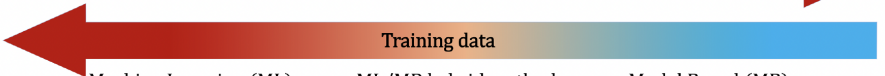
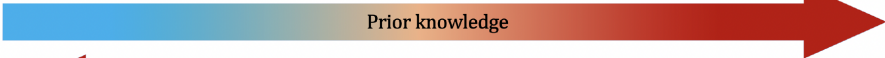
quantum physics



fluid dynamics



Newtonian physics



Machine Learning (ML)
Deep Learning

ML/MB hybrid methods

- Data assimilation
- Physics-informed ML

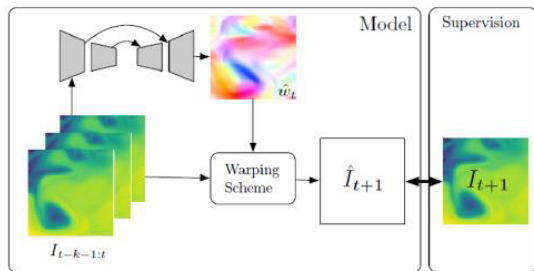
Model Based (MB)
Numerical simulation

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

Incorporating prior information in machine learning

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

- ▶ Loss function regularization, e.g. Physics-Informed Neural Networks (PINNs) [9]
- ▶ Constraints in deep architecture [3, 5]



Advection-diffusion equation

$$\frac{\partial I}{\partial t} + (w \cdot \nabla) I = D \nabla^2 I$$

Warping scheme

$$\hat{I}_{t+1}(x) = \sum_{y \in \Omega} k(x - \hat{w}(x), y) I_t(y)$$

Figure: Advection-diffusion flow [3]

NN architecture to approximate the solution of a linear PDE

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

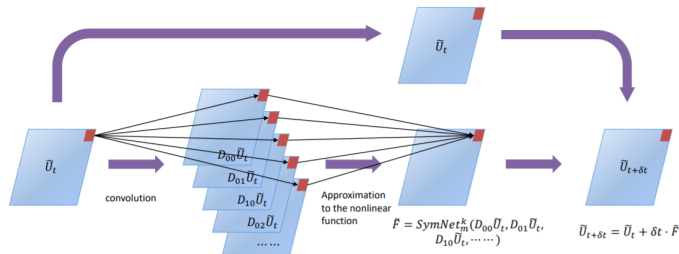


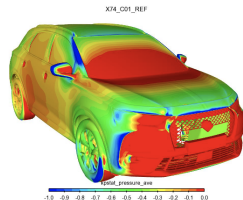
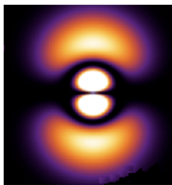
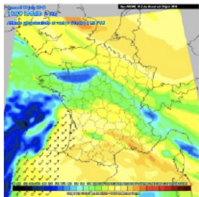
Figure 1: The schematic diagram of a δt -block.

- ▶ Euler discretization: $u(t+1) \approx u(t) + \delta t \cdot F(x, y, u_x(t), \dots) \Rightarrow$ 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



⇒ **Augmenting simplified physical models with data-driven networks**

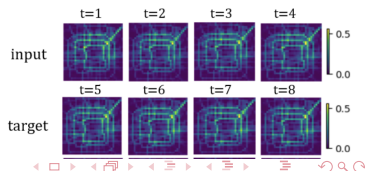
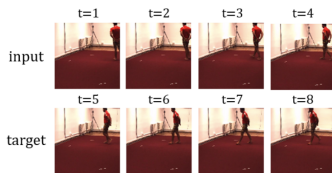
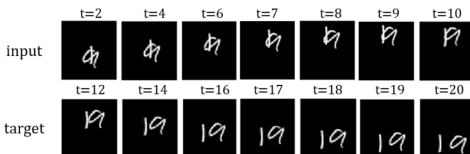
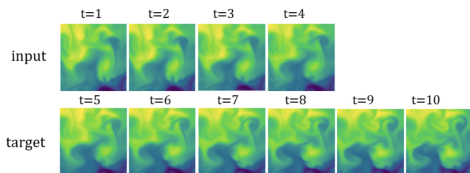
- ▶ Fully vs partial observability: physical models not always applicable in input space
⇒ **Leveraging PDE dynamics in a learned latent space**
- ▶ Proper cooperation between prior physical and data-driven augmentation
⇒ **Decomposition with uniqueness guarantees**

Outline

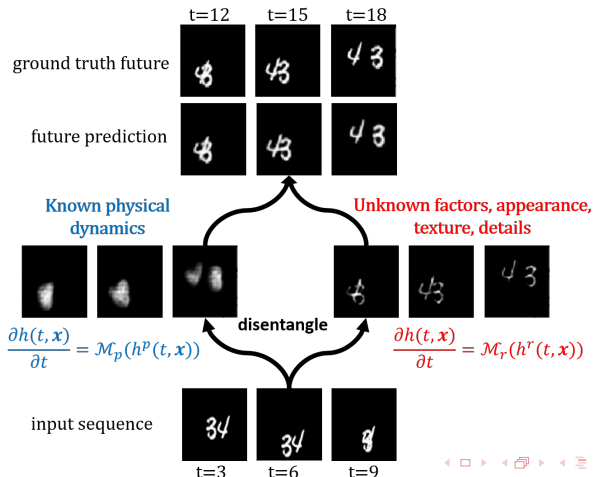
- 1 Context
- 2 **PhyDNet: Physically-constrained deep video prediction**
- 3 APHYNITY: physics & ML cooperation

Unsupervised Video Prediction

- ▶ **Deep learning state-of-the-art for video prediction**
 - ▶ Seminal works: Sequence To Sequence LSTM [11], ConvLSTM [17]
 - ▶ Many further architectures based on 2D/3D CNNs [6, 12, 10], or RNNs [15, 13, 7, 16, 14]
- ▶ **However, still very challenging for pure data-driven methods:**
 - ▶ high-dimensionality of images
 - ▶ inherent uncertainty of future (often blurry predictions)



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



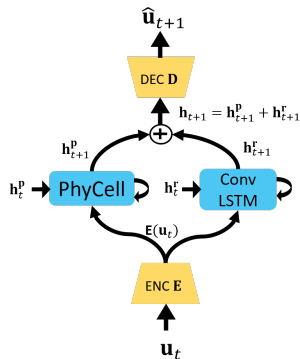
PhyDNet

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} :**

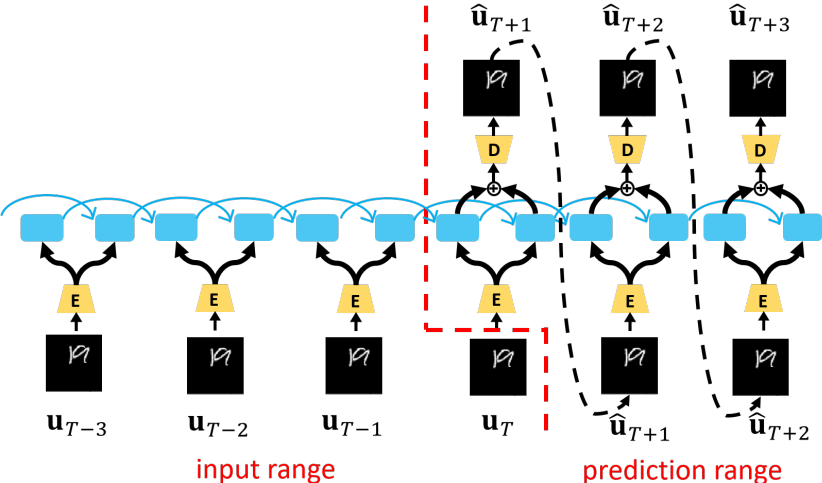
$$\frac{\partial \mathbf{h}(t, \mathbf{x})}{\partial t} = \frac{\partial \mathbf{h}^{\mathbf{P}}}{\partial t} + \frac{\partial \mathbf{h}^{\mathbf{r}}}{\partial t} := \mathcal{M}_p(\mathbf{h}^{\mathbf{P}}, \mathbf{u}) + \mathcal{M}_r(\mathbf{h}^{\mathbf{r}}, \mathbf{u})$$

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



PhyDNet

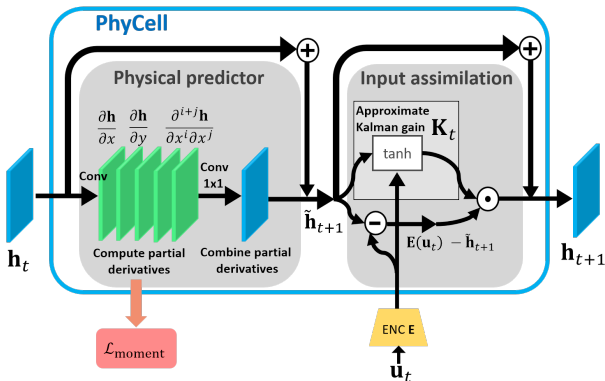
Unrolled PhyDNet architecture



- Prediction/correction: $\mathcal{M}_p(\mathbf{h}, \mathbf{u}) := \Phi(\mathbf{h}) + \mathcal{C}(\mathbf{h}, \mathbf{u})$, discretization:

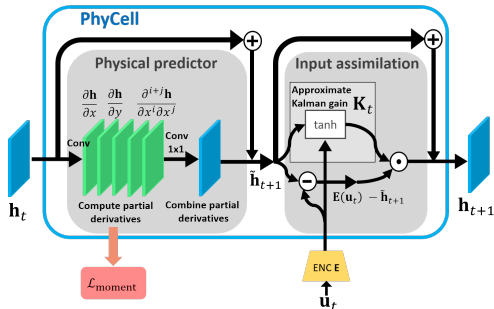
$$\left\{ \begin{array}{l} \tilde{\mathbf{h}}_{t+1} = \mathbf{h}_t + \Phi(\mathbf{h}_t) \quad \text{Prediction} \end{array} \right. \quad (2)$$

$$\left\{ \begin{array}{l} \mathbf{h}_{t+1} = \tilde{\mathbf{h}}_{t+1} + \mathbf{K}_t \odot (\mathbf{E}(\mathbf{u}_t) - \tilde{\mathbf{h}}_{t+1}) \quad \text{Correction} \end{array} \right. \quad (3)$$



PhyCell

Atomic recurrent cell for building physically-constrained RNNs



Prediction-correction revisited with deep learning

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

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

- ▶ $\Phi(\mathbf{h}(t, \mathbf{x})) = \sum_{i,j:i+j \leq q} c_{i,j} \frac{\partial^{i+j} \mathbf{h}}{\partial x^i \partial y^j} (t, \mathbf{x})$,
PDE-Net in latent space

- ▶ Approximate $\frac{\partial^{i+j} \mathbf{h}}{\partial x^i \partial y^j}$ with conv & moment loss

- ▶ $\{c_{i,j}\}$ learned

Correction:

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

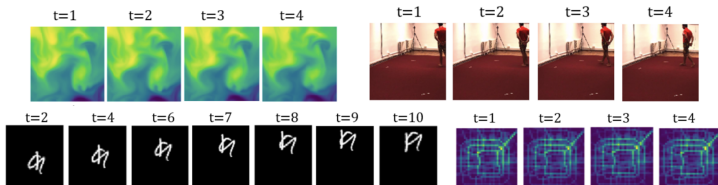
- ▶ Learned Kalman gain \mathbf{K}_t for correction: $\kappa_t = \tanh(\mathbf{W}_h * \tilde{\mathbf{h}}_{t+1} + \mathbf{W}_u * \mathbf{E}(\mathbf{u}_t) + \mathbf{b}_k)$
- ▶ \neq locally linear approximate Kalman gain in [1]

State-of-the-art wrt

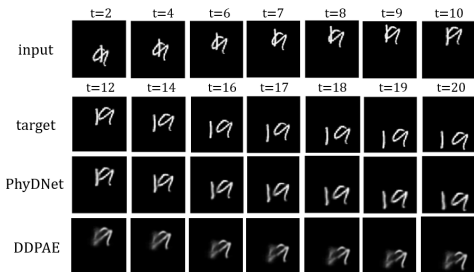
- ▶ Recent general video prediction methods
- ▶ specialized models for Moving Mnist and SST

Method	Moving MNIST			Traffic BJ			Sea Surface Temperature			Human 3.6		
	MSE	MAE	SSIM	MSE $\times 100$	MAE	SSIM	MSE $\times 10$	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

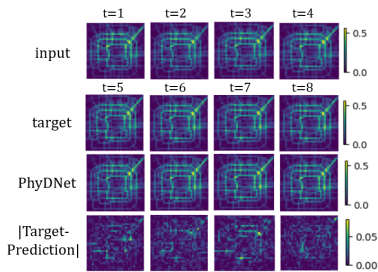
Table 1. Quantitative forecasting results of PhyDNet compared to baselines using various datasets. Numbers are copied from original or citing papers. * corresponds to results obtained by running online code from the authors. The first five baseline are general deep models applicable to all datasets, 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.



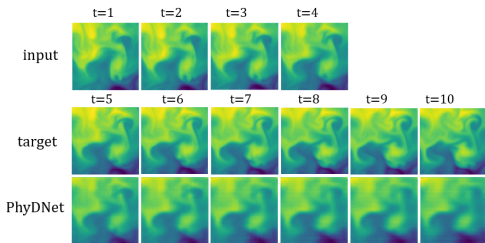
Qualitative results



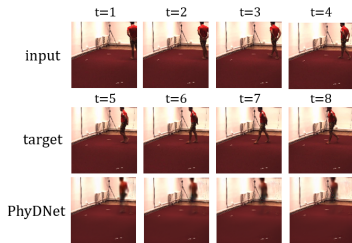
(a) Moving MNIST



(b) Traffic BJ



(c) Sea Surface Temperature

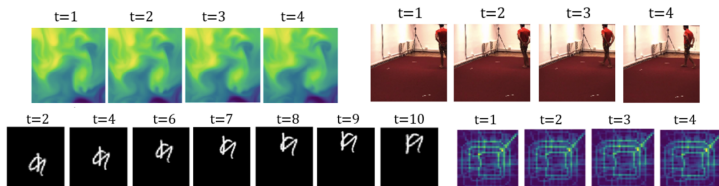


(d) Human 3.6

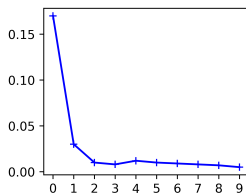
Ablation study

Method	Moving MNIST			Traffic BJ			Sea Surface Temperature			Human 3.6		
	MSE	MAE	SSIM	MSE \times 100	MAE	SSIM	MSE \times 10	MAE	SSIM	MSE / 10	MAE / 100	SSIM
ConvLSTM	103.3	182.9	0.707	48.5*	17.7*	0.978*	45.6*	63.1*	0.949*	50.4*	18.9*	0.776*
PhyCell	50.8	129.3	0.870	48.9	17.9	0.978	38.2	60.2	0.969	42.5	18.3	0.891
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

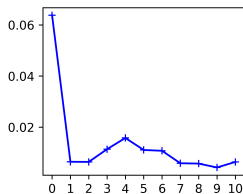
Table 2. An ablation study shows the consistent performance gain on all datasets of our physically-constrained PhyCell vs the general purpose ConvLSTM, and the additional gain brought up by the disentangling architecture PhyDNet. * corresponds to results obtained by running online code from the authors.



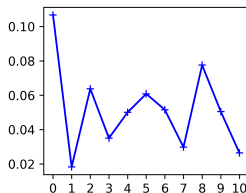
Analysis of learned filters



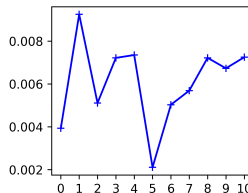
Moving MNIST



Traffic BJ



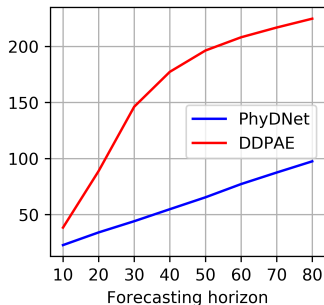
SST



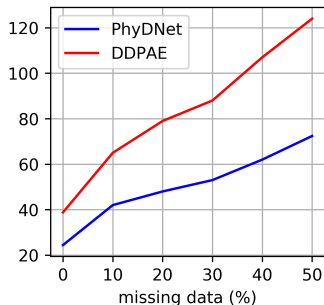
Human 3.6

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

Long-term prediction and missing data



(a) Long-term forecasting



(b) Missing data

Figure: MSE comparison between PhyDNet and DDPAE [4] when dealing with unreliable inputs.

Outline

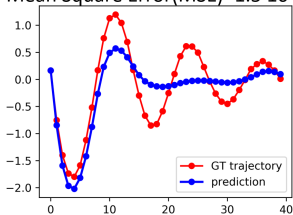
- 1 Context
- 2 PhyDNet: Physically-constrained deep video prediction
- 3 APHYNITY: physics & ML cooperation**

Motivation: data-driven vs. simplified physical models

- ▶ **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

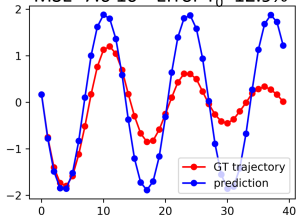
$$\text{Damped pendulum: } \frac{d^2\theta}{dt^2} + \omega_0^2 \sin \theta + \lambda \frac{d\theta}{dt} = 0$$

Mean Square Error(MSE)= $1.5 \cdot 10^{-1}$



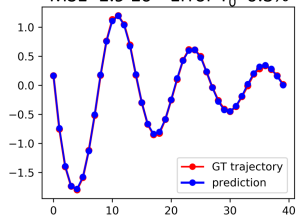
(a) Data-driven Neural ODE

MSE= $7.6 \cdot 10^{-1}$ Error $T_0=12.9\%$



(b) Simple physical model

MSE= $1.9 \cdot 10^{-4}$ Error $T_0=0.3\%$



(c) Our APHYNITY framework

⇒ **Augmenting PHYSical models for ideNtIfying and forecasTing complex dYnamic (APHYNITY)**

APHYNITY

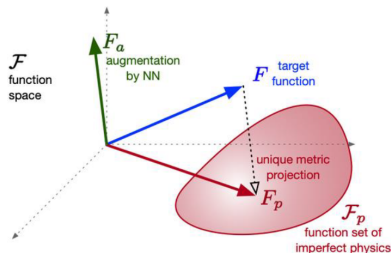
- ▶ $\frac{dX_t}{dt} = F(X_t)$, $X_t \in \mathbb{R}^d$ (vector) or $X_t(\mathbf{x}) \in \mathbb{R}^d$, $\mathbf{x} \in \Omega \subset \mathbb{R}^k$ (vector field)
 - ▶ $F \in \mathcal{F}$ normed vector space, $F_p \in \mathcal{F}_p \subset \mathcal{F}$ physical model (ODE/PDE)
- ▶ **Augment approximate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

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

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

$$\min_{F_p \in \mathcal{F}_p, F_a \in \mathcal{F}} \|F_a\| \text{ subject to } F = (F_p + F_a) \quad (4)$$

- ▶ If \mathcal{F}_p Chebyshev set¹, decomposition in Eq (4) 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

¹In finite-dim space, closed convex sets

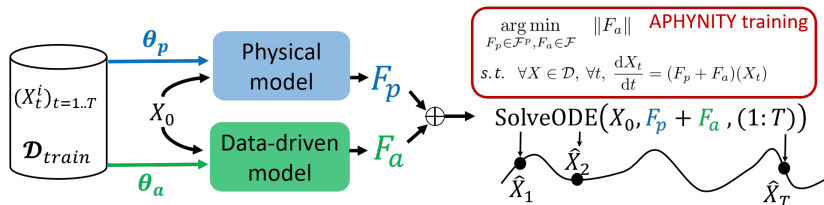
APHYNITY training

- Dataset of observed trajectories: $\mathcal{D} = \{X : [0, T] \rightarrow \mathcal{F} \mid \forall t \in [0, T], \frac{dX_t}{dt} = F(X_t)\}$

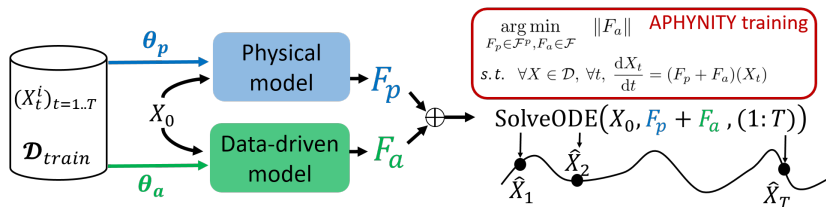
APHYNITY objective:

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

- 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):

$$\mathcal{L}_{\lambda_j}(\theta_p, \theta_a) = \|F_a^{\theta_a}\| + \lambda_j \cdot \mathcal{L}_{traj}(\theta_p, \theta_a) \quad (5)$$

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

- ▶ $\theta = (\theta_p, \theta_a)$, 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

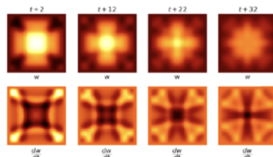
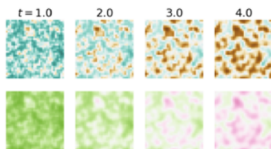
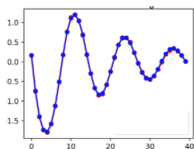
Initialization: $\lambda_0 \geq 0, \tau_1 > 0, \tau_2 > 0$;
for *epoch* = 1 : N_{epochs} **do**
 for *iter* in 1 : N_{iter} **do**
 for *batch* in 1 : B **do**
 $\theta_{j+1} = \theta_j - \tau_1 \nabla [\lambda_j \mathcal{L}_{traj}(\theta_j) + \|F_a\|]$
 $\lambda_{j+1} = \lambda_j + \tau_2 \mathcal{L}_{traj}(\theta_{j+1})$

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

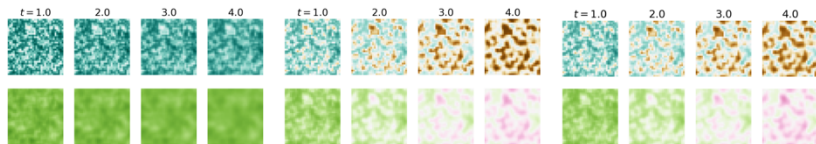


Experiments: APHYNITY results

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		

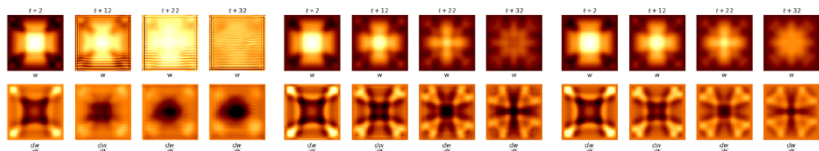
- ▶ Better forecasting performances
- ▶ Better physical parameter identification
- ▶ $\|F_a\|^2 \sim$ level of F_p approximation

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) Ground truth simulation

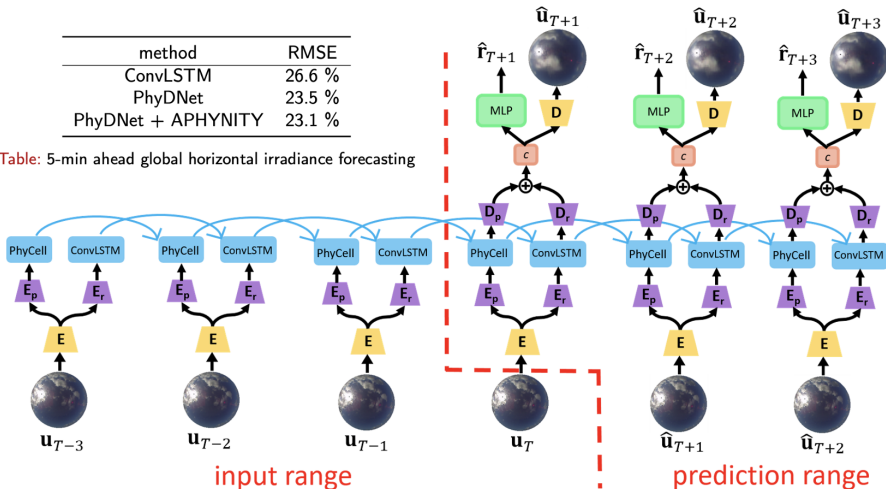
Figure 4: Comparison between the prediction of APHYNITY when c is estimated and Neural ODE for the damped wave equation. Note that $t + 32$ is already beyond the dataset horizon ($t + 25$), showing the consistency of APHYNITY method.

Application to solar energy forecasting (CVPR'20 workshop)

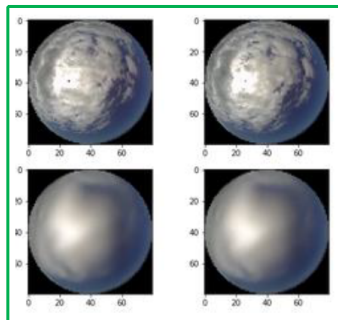
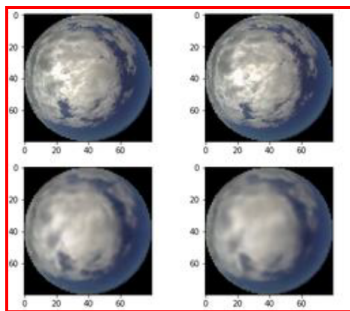
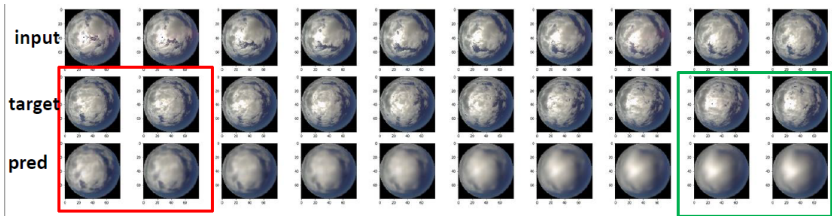
- ▶ Short-term (<20min) solar irradiance forecasting with fisheye images
- ▶ Improved PhyDNet model with separate encoders/decoders & $\min \|F_a\|^2$

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, control (model-based RL), computational fluid dynamics (e.g. modeling turbulences)
 - ▶ Others...
- ▶ Beyond summing derivatives: $F = F_p + F_a$

Thank you for your attention!

Questions?

- ▶ **PhyDNet: Vincent Le Guen, Nicolas Thome**
 - ▶ **CVPR'20 paper:** Disentangling Physical Dynamics from Unknown Factors for Unsupervised Video Prediction
 - ▶ **CVPR'20 OMNI-CV workshop:** A Deep Physical Model for Solar Irradiance Forecasting with Fisheye Images.
 - ▶ **GitHub code:** <https://github.com/vincent-leguen/PhyDNet>
- ▶ **APHYNITY: Yuan Yin, Vincent Le Guen, Jérémie Dona, Ibrahim Ayed, Emmanuel de Bézenac, Nicolas Thome, Patrick Gallinari.**
 - ▶ **ICLR'21 paper:** Augmenting Physical Models with Deep Networks for Complex Dynamics Forecasting



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] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li F Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 517–526, 2018.
- [5] 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.
- [6] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *International Conference on Learning Representations (ICLR)*, 2015.
- [7] Marc Oliu, Javier Selva, and Sergio Escalera. Folded recurrent neural networks for future video prediction. In *European Conference on Computer Vision (ECCV)*, pages 716–731, 2018.
- [8] 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.

References II

- [9] 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.
- [10] Fitsum A Reda, Guilin Liu, Kevin J Shih, Robert Kirby, Jon Barker, David Tarjan, Andrew Tao, and Bryan Catanzaro.
SDC-Net: Video prediction using spatially-displaced convolution.
In *European Conference on Computer Vision (ECCV)*, pages 718–733, 2018.
- [11] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov.
Unsupervised learning of video representations using LSTMs.
In *International Conference on Machine Learning (ICML)*, pages 843–852, 2015.
- [12] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba.
Generating videos with scene dynamics.
In *Advances In Neural Information Processing Systems (NeurIPS)*, pages 613–621, 2016.
- [13] Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and Philip S Yu.
PredRNN++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning.
arXiv preprint arXiv:1804.06300, 2018.
- [14] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei.
Eidetic 3D LSTM: A model for video prediction and beyond.
In *International Conference on Learning Representations (ICLR)*, 2019.
- [15] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and S Yu Philip.
PredRNN: Recurrent neural networks for predictive learning using spatiotemporal lstms.
In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 879–888, 2017.
- [16] Yunbo Wang, Jianjin Zhang, Hongyu Zhu, Mingsheng Long, Jianmin Wang, and Philip S Yu.
Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics.
In *Computer Vision and Pattern Recognition (CVPR)*, pages 9154–9162, 2019.

References III

- [17] Shi Xingjian, Zhouong 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.