Deep time series forecasting with prior knowledge

Nicolas Thome - Prof. at Sorbonne University
ISIR Lab, Machine Learning Team (MLIA)

Time series and transfer learning workshop

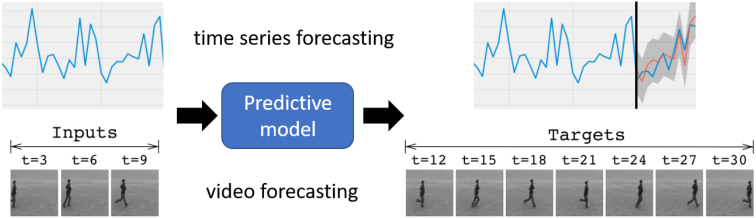
https://gtsbrain-paris.github.io/events.html

19th of October, Paris

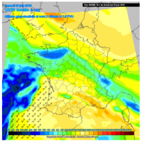
Outline

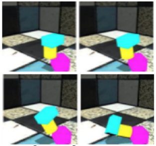
Spatio-temporal forecasting

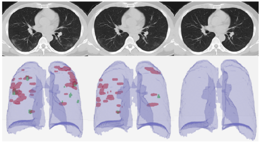
- **Future prediction** of time series with complex temporal and potentially spatial correlations

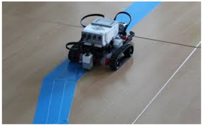


- **Crucial for many applications:** weather and climate science, healthcare, robotics, finance, *etc*


weather forecasts

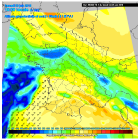

physical reasoning


medical prognosis , *e.g.* COVID evolution

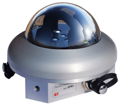

robot visual navigation

Spatio-temporal forecasting & big data

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

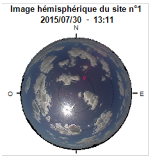


weather forecasts

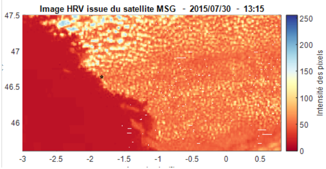


sensors, pyranometers

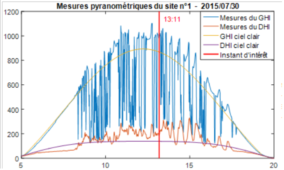
monitoring cameras



fisheye images



satellite images

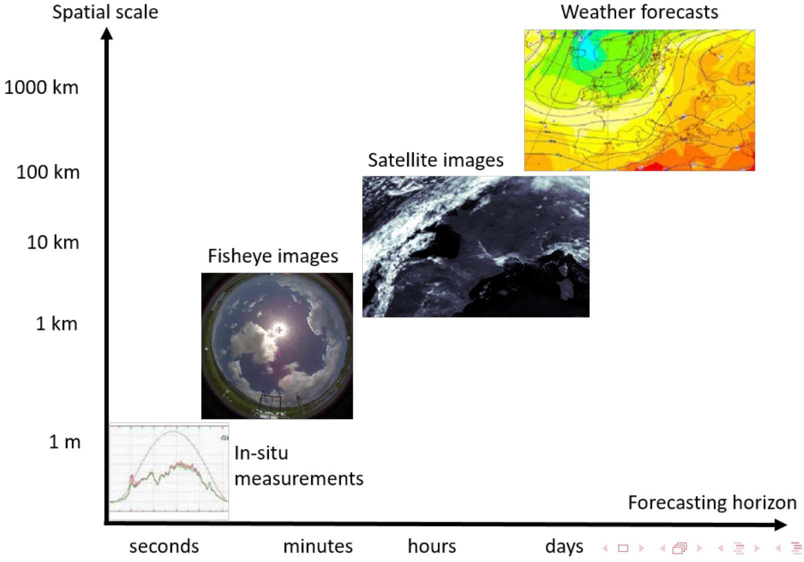


time series

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

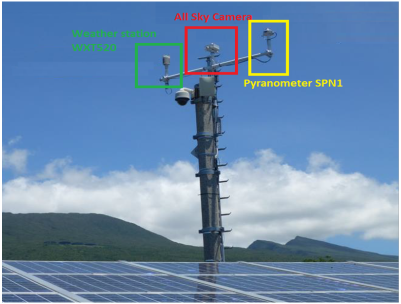
Solar irradiance forecasting

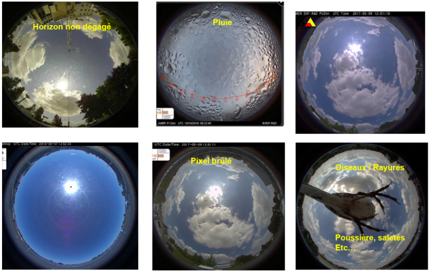
V. Le Guen's PhD ('18-'21): Industrial application at EDF



Solar irradiance forecasting with sky images

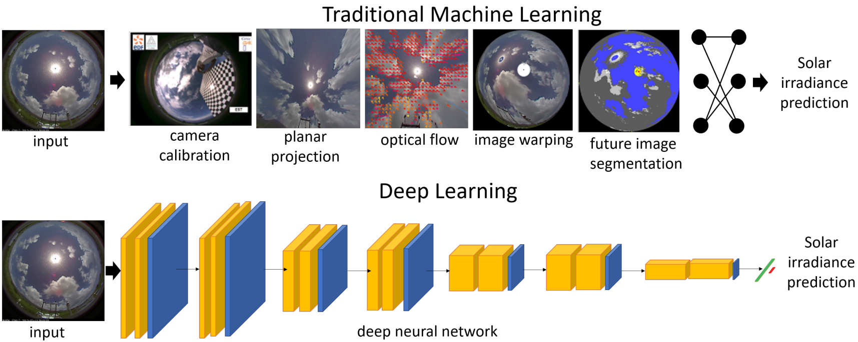
Data: > 7 million images and measured solar irradiance every 10s





Goal: predict future solar irradiance values (0-20min) given previous fisheye images

Machine Learning and Deep Learning



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

Pure data-driven forecasts struggle to properly extrapolate:

- lag behind the ground truth
- do not capture sharp patterns (blurry predicted trajectory)

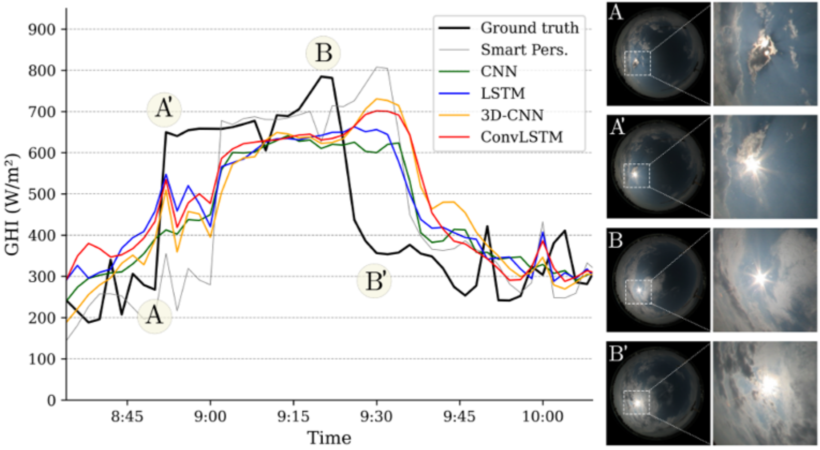


Figure: 5min solar irradiance forescasting, from [13]

Research directions to improve forecasts

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

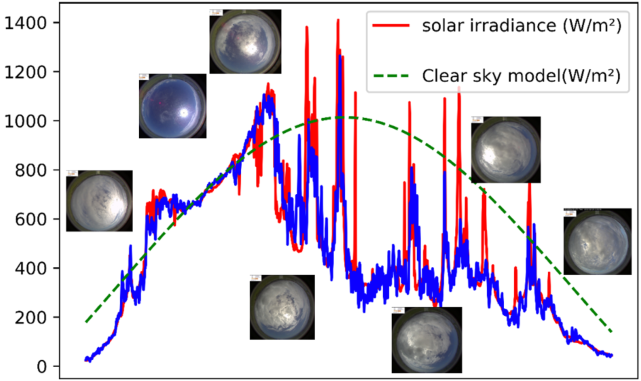
1. Prior knowledge in training loss function
2. Physical knowledge in forecasting model

Outline

Time series forecasting

- **Long horizon forecasts**, *i.e.* multi-step setting
- **Non-stationary** time series, that can present abrupt changes

Important in many contexts, e.g. electricity (anticipate future drops of production), etc...



Traditional methods:

- Auto-Regressive models (ARMA, ARIMA,...) [1]
- State Space Models (Exponential smoothing, ...) [9]

– Assumption: stationary time series

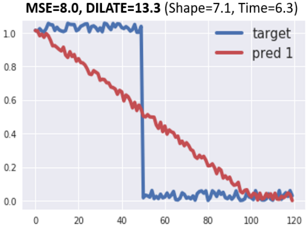
Deep learning models:

- Seq2Seq Recurrent Neural Networks [19]
- Complex architectures for multivariate forecasting: attention mechanisms, tensor factorizations [18]
- Deep State Space Models for modeling uncertainty [15]

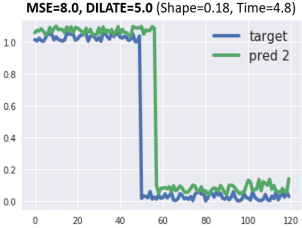
... but those models trained with the Mean Squared Error (MSE) !

Motivation: MSE Loss Limitation

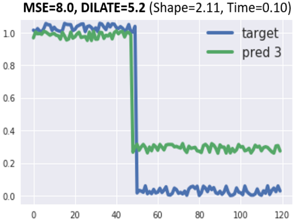
- MSE loss typically used for training forecasting problems not adapted to judge the quality of a forecast.



Non informative prediction

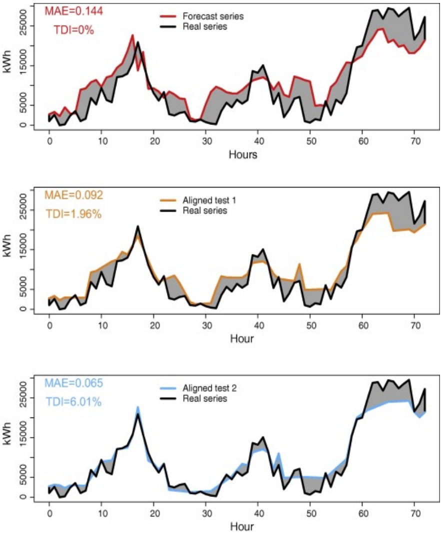


Correct shape, time delay



Correct time, inaccurate shape

Specific Metric for time series forecasting



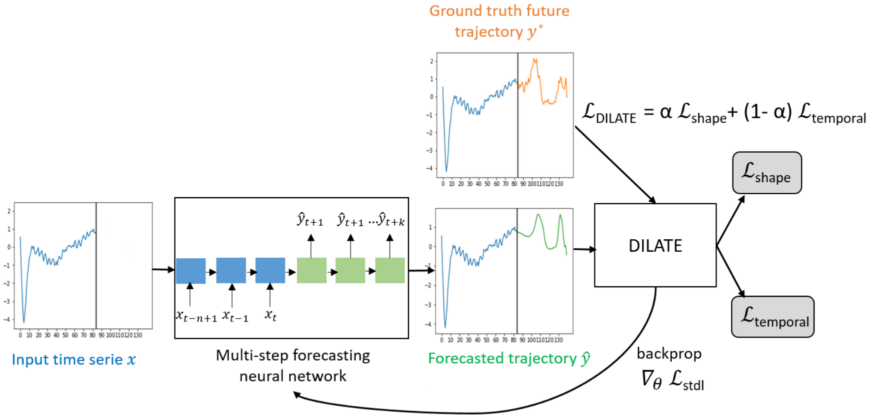
- Change Point Detection [2, 10]
- Hausdorff distance [8, 16]
- Ramp score [6, 17]
- Time Distrosion Index (TDI) [7]

... but not differentiable! How to train deep models?

DILATE (DIstortion Loss with shApe and TimE)

- Training dataset: N input time series $\mathcal{A} = \{\mathbf{x}_i\}_{i \in \{1:N\}}$
  - $\mathbf{x}_i = (\mathbf{x}_i^1, ..., \mathbf{x}_i^n) \in \mathbb{R}^{p \times n}$ input of length n
  - $\overset{*}{\mathbf{y}}_i = (\overset{*}{\mathbf{y}}_i{}^1, ..., \overset{*}{\mathbf{y}}_i{}^k)$ GT output of length k
  - $\hat{\mathbf{y}}_i = (\hat{\mathbf{y}}_i^1, ..., \hat{\mathbf{y}}_i^k) \in \mathbb{R}^{d \times k}$ predicted output of length k

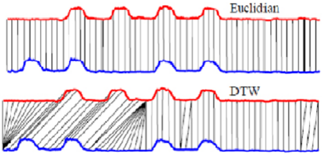
$$\mathcal{L}_{DILATE}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) = \alpha \; \mathcal{L}_{shape}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) + (1 - \alpha) \; \mathcal{L}_{temporal}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) \quad (1)$$



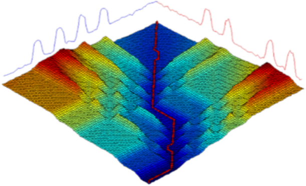
DILATE - shape loss

- Based on dynamic time warping (DTW) that computes the optimal alignment $\mathbf{A}^*$ between 2 time series:

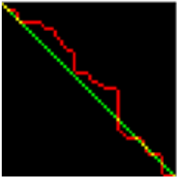
$$DTW(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) = \min_{\mathbf{A} \in \mathcal{A}_{k,k}} \left\langle \mathbf{A}, \boldsymbol{\Delta}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) \right\rangle$$



MSE *vs* DTW loss



Pairwise cost matrix Δ



Optimal alignment $\mathbf{A}^*$

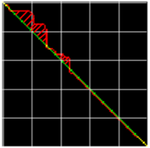
- Soft-DTW [4]: soft minimum to make DTW differentiable

$$\mathcal{L}_{shape}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) = DTW_\gamma(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) := -\gamma \log \left(\sum_{\mathbf{A} \in \mathcal{A}_{n,m}} \exp \left(-\frac{\left\langle \mathbf{A}, \boldsymbol{\Delta}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) \right\rangle}{\gamma} \right) \right)$$

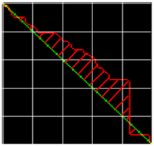
DILATE - temporal loss

- Quantify the deviation of optimal path $\mathbf{A}^*$ from the main diagonal with the Time Distortion Index (TDI) [7]

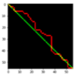
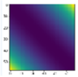
Small red area:
⇒ small temporal distortion



Large red area:
⇒ large temporal distortion



$$TDI(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) = \langle \mathbf{A}^*, \mathbf{\Omega} \rangle = \left\langle \underset{\mathbf{A} \in \mathcal{A}_{k,k}}{\arg\min} \left\langle \mathbf{A}, \mathbf{\Delta}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) \right\rangle, \mathbf{\Omega} \right\rangle = \langle$$

 ,  $\rangle$

- We introduce a smooth relaxation of the TDI:

$$\mathcal{L}_{temporal}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) := \left\langle \mathbf{A}^*_\gamma, \mathbf{\Omega} \right\rangle = \frac{1}{Z} \sum_{\mathbf{A} \in \mathcal{A}_{n,m}} \langle \mathbf{A}, \mathbf{\Omega} \rangle \exp^{-\frac{\left\langle \mathbf{A}, \mathbf{\Delta}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) \right\rangle}{\gamma}}$$

DILATE experiments

- **Experimental setup:** evaluate the k-step future trajectories
- **Non stationary datasets:** Synthetic, ECG5000, Traffic, ETTH1 (horizon 96)
- **DILATE training:** equivalent results evaluated on MSE, better results evaluated on shape and time metrics

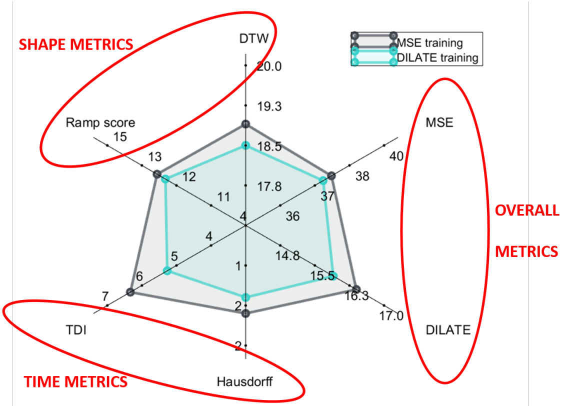
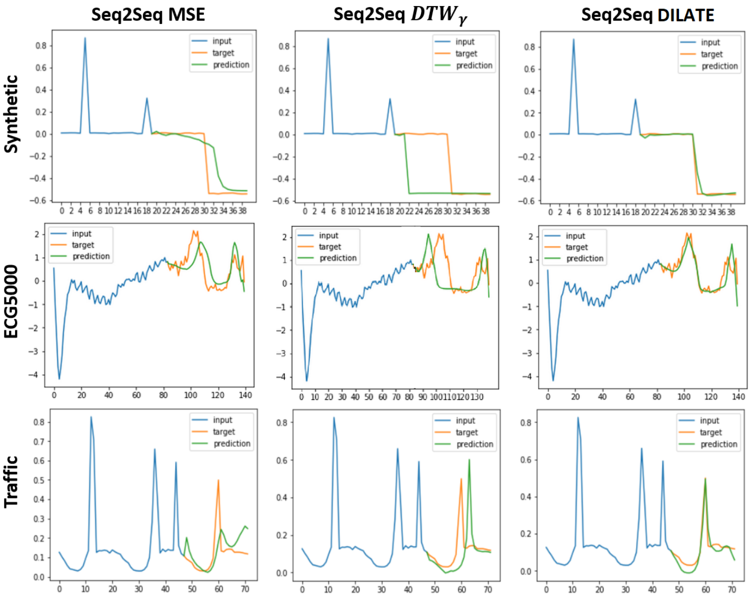


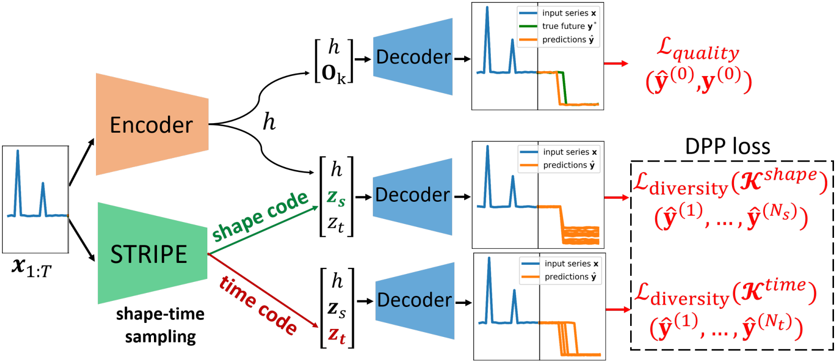
Figure: Results on `Electricity` with the Informer model [20].

Qualitative forecasting results



Extension to probabilistic forecasting

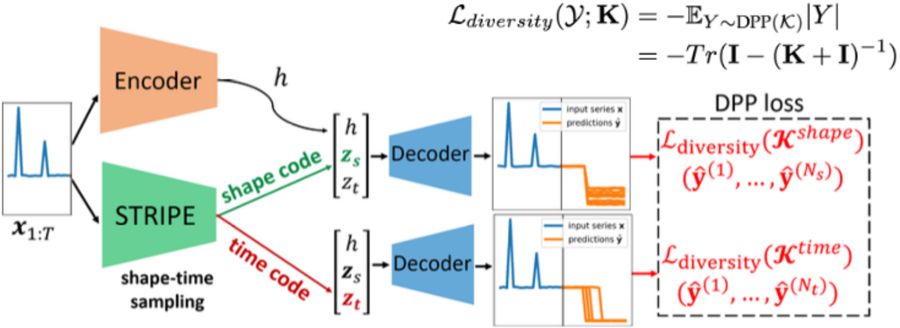
- **Goal:** produce a small set of sharp and diverse future trajectories
- **STRIPE:** Shape and Time diverRsIty in Probabilistic forEcasting



1. Train a quality loss for deterministic forecasting (encoder, decoder)
2. Diversification with a determinantal point processes (DPP) with shape and time criteria (encoder, decoder frozen)

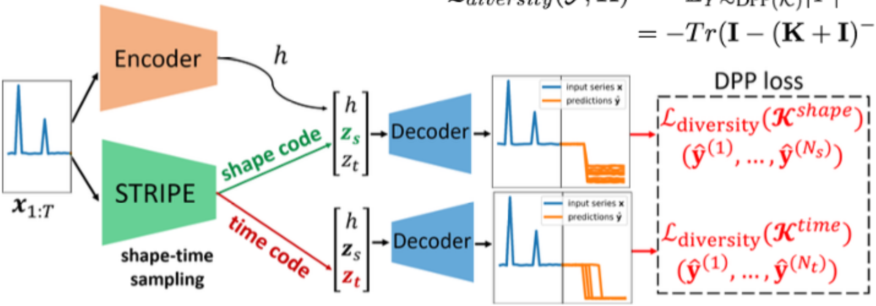
SRTIPE diversification: DPP

- Train STRIPE diversifier with a DPP loss $\mathcal{L}_{diversity}(\mathcal{Y}; \mathbf{K})$



- DPP: kernel diversification ⇒ shape and time kernels!

$$\mathcal{L}_{diversity}(\mathcal{Y};\mathbf{K}) = -\mathbb{E}_{Y\sim \mathrm{DPP}(\mathcal{K})}|Y| = -Tr(\mathbf{I}-(\mathbf{K}+\mathbf{I})^{-1})$$



- **Deriving valid PSD kernels from shape and time criteria**

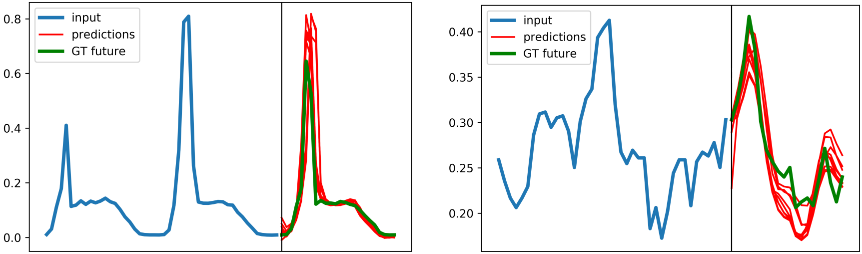
criterion	differentiable loss	PSD sim. kernel
shape	$\mathrm{DTW}_\gamma^{\mathbf{\Delta}}(\mathbf{y},\mathbf{z})$	$e^{-\mathrm{DTW}_\gamma^{\mathbf{\Delta}}(\mathbf{y},\mathbf{z})/\gamma}$
time	$\mathrm{TDI}_\gamma^{\mathbf{\Delta},\mathbf{\Omega}_{\text{dissim}}}(\mathbf{y},\mathbf{z})$	$e^{-\mathrm{DTW}_\gamma^{\mathbf{\Delta}}(\mathbf{y},\mathbf{z})/\gamma} \times \mathrm{TDI}_\gamma^{\mathbf{\Delta},\mathbf{\Omega}_{\text{sim}}}(\mathbf{y},\mathbf{z})$

STRIPE results

- Take home message: large increase in diversity without scarifying quality

Methods	$H_{quality}(\cdot)$ (↓) DTW	TDI	DILATE	$H_{diversity}(\cdot)$ (↓) DTW	TDI	DILATE	$F1$ score (↓) DTW	TDI	DILATE	CRPS (↓)
DeepAR [15]	42.9 ± 6.6	16.6 ± 7.6	33.5 ± 6.0	23.9 ± 3.5	**12.8 ± 2.5**	22.7 ± 2.2	30.7	14.5	27.1	62.4 ± 9.9
cVAE DILATE	**11.7 ± 1.5**	9.4 ± 2.2	**14.2 ± 1.5**	18.8 ± 1.3	48.6 ± 2.2	33.9 ± 3.9	14.4	15.7	20.0	62.2 ± 4.2
variety loss [61] DILATE	15.6 ± 3.4	10.2 ± 1.1	16.8 ± 0.9	22.7 ± 4.1	37.7 ± 4.9	30.8 ± 1.0	18.5	16.1	21.7	62.6 ± 3.0
entropy reg. [62] DILATE	13.8 ± 3.1	8.8 ± 2.2	**15.0 ± 1.6**	20.4 ± 2.8	42.0 ± 7.8	32.6 ± 2.3	16.5	14.5	20.5	62.4 ± 3.9
Diverse DPP [1] DILATE	**12.9 ± 1.2**	9.8 ± 2.1	15.1 ± 1.5	18.6 ± 1.6	42.8 ± 10.1	31.3 ± 5.7	15.2	15.9	20.4	60.7 ± 1.6
GDPP [68] DILATE	14.8 ± 2.9	11.7 ± 8.4	**14.4 ± 2.1**	20.8 ± 2.4	25.2 ± 7.2	23.9 ± 4.5	17.3	15.9	17.9	63.4 ± 6.4
STRIPE [25]	16.8 ± 0.5	**6.7 ± 0.4**	15.4 ± 0.5	16.1 ± 1.1	**13.2 ± 1.7**	**17.7 ± 0.6**	16.4	**8.8**	16.5	60.5 ± 0.4
STRIPE++	13.5 ± 0.5	9.2 ± 0.5	**15.0 ± 0.3**	**12.9 ± 0.3**	16.3 ± 1.2	**17.9 ± 0.6**	**13.2**	11.7	**16.3**	**48.6 ± 0.6**

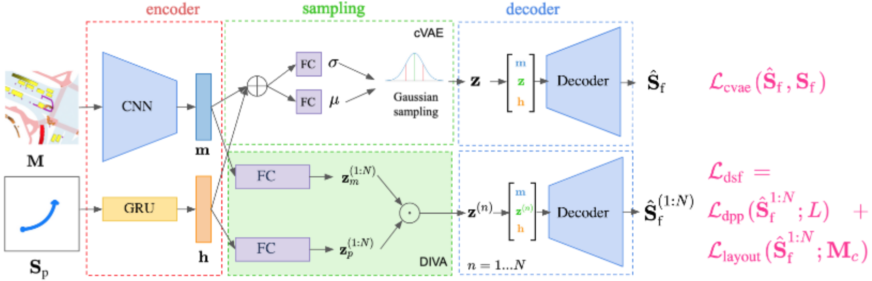
- Qualitative predictions:



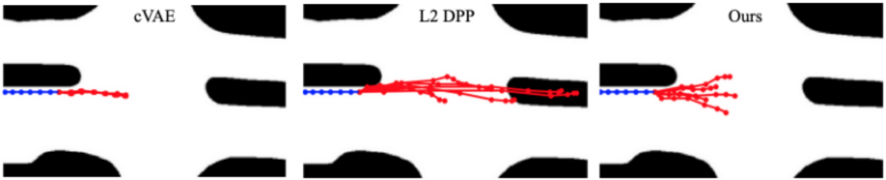
(a) Traffic (b) Electricity

Application to autonomous driving

- DIVA: DIVerse trajectory prediction with Admissibility constraints



  - Layout loss for fulfilling domain contraints (predictions in drivable area)



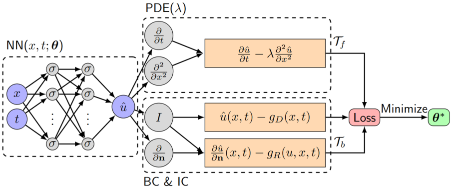
Outline

Related work: incorporating physical knowledge in ML

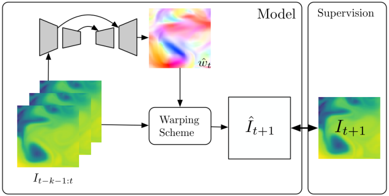
Hybrid models combining MB and ML (gray box)

Loss function regularization

Constraints in deep architectures



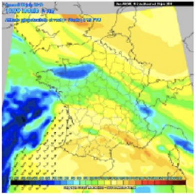
Physics-informed neural networks [14]

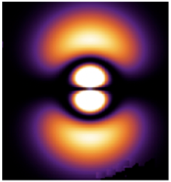


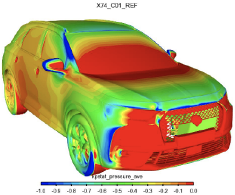
Advection-diffusion model [5]

Focus: simplified physical models

Physical models: often approximations of real-world dynamics



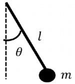




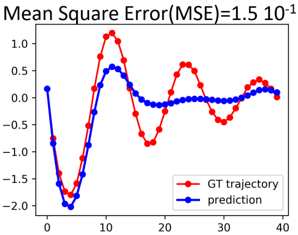
- A complete description of a complex natural phenomenon is out of reach, *e.g.* climate, earth modelling
- Approximations are made to make the numerical resolution tractable, *e.g.* reduced-order models, resolution on coarse meshes

Motivation: data-driven vs. simplified physical models

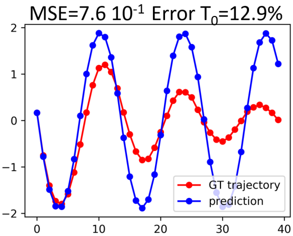
Damped pendulum: $\frac{d^2\theta}{dt^2} + \omega_0^2 \sin\theta + \lambda \frac{d\theta}{dt} = 0$



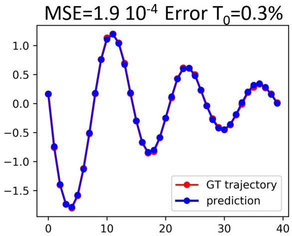
- **Data-driven models** struggle to extrapolate complex dynamics, in particular in data-scarce contexts
- **Physical models** fail to extrapolate when they are misspecified: forecasting & parameter identification failure



(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

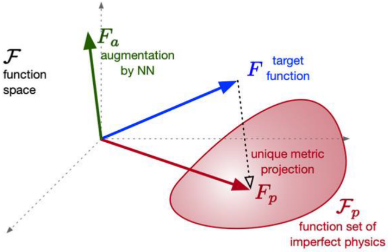
- $\frac{\mathrm{d}X_t}{\mathrm{d}t} = F(X_t)$, $X_t \in \mathbb{R}^d$ (vector) or $X_t(\mathbf{x}) \in \mathbb{R}^d$, $\mathbf{x} \in \Omega \subset \mathbb{R}^k$ (vector field)
  - $F \in \mathcal{F}$ normed vector space, $F_p \in \mathcal{F}_p \subset \mathcal{F}$ physical model (ODE/PDE)
- **Augment 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[1], decomposition in Eq (2) exists and is unique (metric projection onto $\mathcal{F}_p$).



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

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

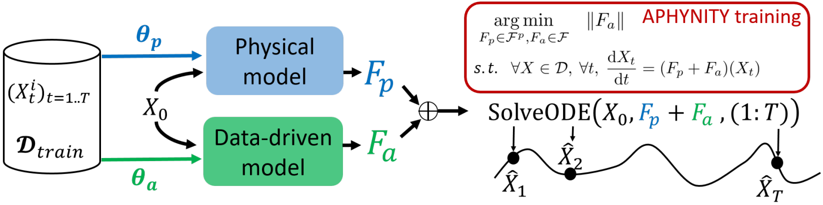
APHYNITY training

- 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:

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

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

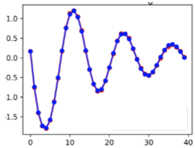


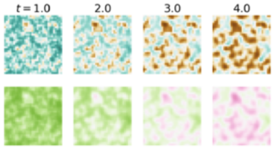
APHYNITY - quantitative results

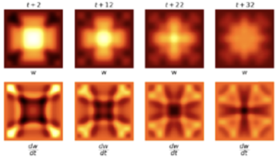
Experiments on 3 classes of physical phenomena:

- **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





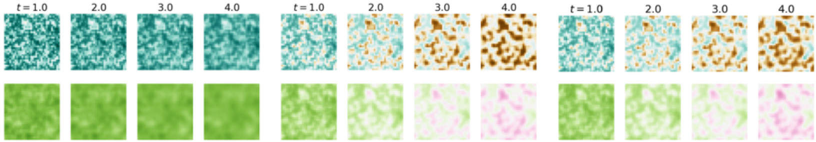


Experiments: APHYNITY results

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

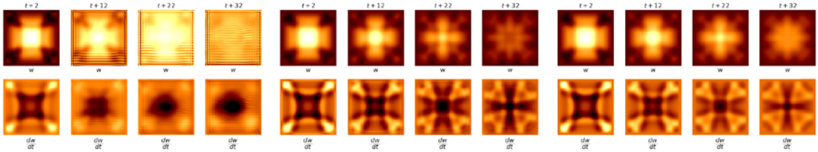
- Better forecasting performances
- Better physical parameter identification
- $\|\|F_a\|\|^2 \sim$ level of F_p approximation

APHYNITY - qualitative results



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

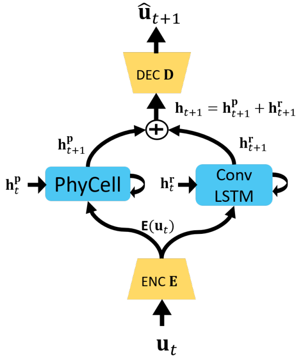
Figure 3: Comparison of predictions of two components u (top) and v (bottom) of the reaction-diffusion system. Note that $t = 4$ is largely beyond the dataset horizon ($t = 2.5$).

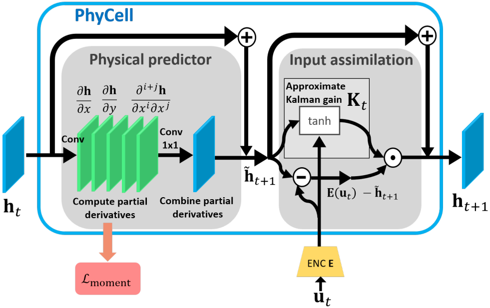


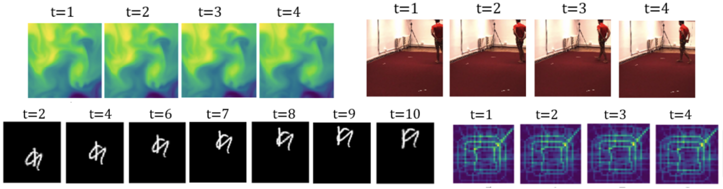
(a) Neural ODE (b) APHYNITY Param PDE (c) (c) Ground truth simulation

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 Video Prediction: PhyDNet (CVPR'20)





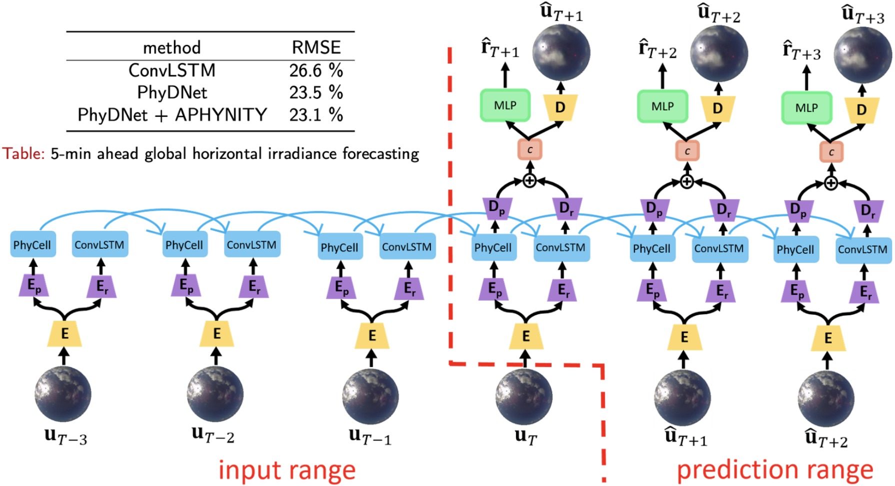


Application to solar energy forecasting (CVPR'20 workshop)

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

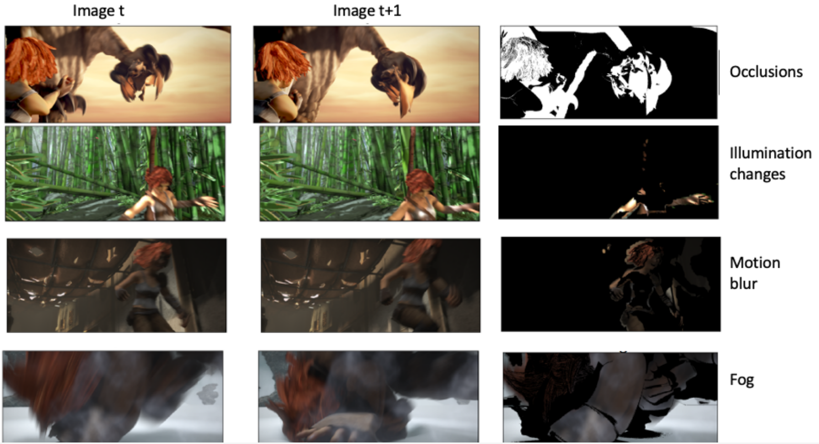
method	RMSE
ConvLSTM	26.6 %
PhyDNet	23.5 %
PhyDNet + APHYNITY	23.1 %

Table: 5-min ahead global horizontal irradiance forecasting



Application to optical flow

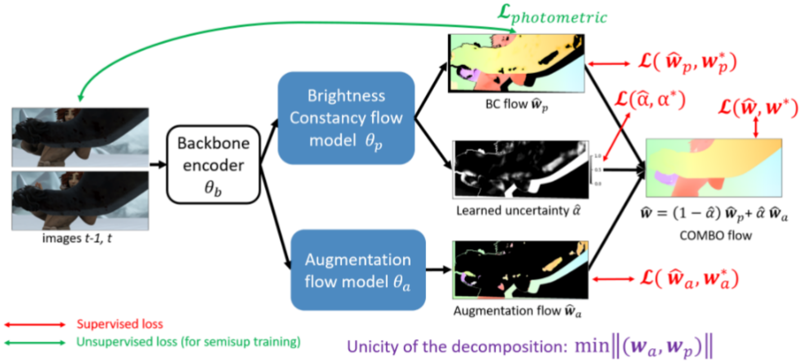
- Deep learning models: trained with complex curriculum, *i.e.* synthetic data (Chairs, Things, Sintel), real data (HD1K, Kitti)
- Traditional methods: based on brightness consistency (BC) assumption: $\frac{\partial I}{\partial t}(t, \mathbf{x}) + \mathbf{w}(t, \mathbf{x}) \cdot \nabla I(t, \mathbf{x})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



COMBO model for optical flow (ECCV'22)

- 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: (3)

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

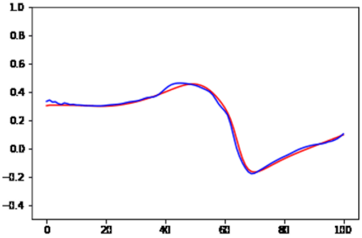


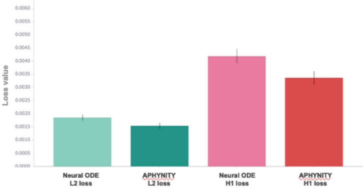
Unicity of the decomposition: $\min \|(w_a, w_p)\|$

Hybrid models for computation fluid dynamics (CFD)

- **Learning dynamics for 1D CFD equations (Burgers):**
  - $\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} - u \frac{\partial u}{\partial x}$, discretization:
  - $\frac{u^{n+1} - u^n}{\Delta t} = \nu \frac{\partial^2 u^{n+1}}{\partial x^2} - u^n \frac{\partial u^{n+1}}{\partial x}$

  $\Rightarrow$ Simulating $\frac{\partial^2 u^{n+1}}{\partial x^2}$ (diffusion), learning $u^n \frac{\partial u^{n+1}}{\partial x}$ (non-linear transport)
- **Goal for the proof of concept:**
  - **Good prediction performances** ($\sim$ full simulation), good physical estimation (ν)
  - **Speeding up computation** (non linear system inversion for learned terms)
  - **Improved training with H_1 loss**





Questions?

Loss function regularization

- **T-PAMI'23 paper**
- **DILATE: NeurIPS'19, GitHub:** https://github.com/vincent-leguen/DILATE
- **STRIPE: NeurIPS'20, GitHub:** https://github.com/vincent-leguen/STRIPE
- **DIVA:, ICPR'22**

Augmented physical models:

- **PhyDNet: CVPR'20, CVPR'20 workshop**
  GitHub: https://github.com/vincent-leguen/PhyDNet
- **APHYNITY: ICLR'21, GitHub:** https://github.com/yuan-yin/aphynity
- **COMBO: ECCV'22, GitHub:** https://github.com/vincent-leguen/COMBO

References I

[1] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung.
Time series analysis: forecasting and control.
John Wiley & Sons, 2015.

[2] Wei-Cheng Chang, Chun-Liang Li, Yiming Yang, and Barnabás Póczos.
Kernel change-point detection with auxiliary deep generative models.
In *International Conference on Learning Representations (ICLR)*, 2019.

[3] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud.
Neural ordinary differential equations.
In *Advances in neural information processing systems (NeurIPS)*, pages 6571–6583, 2018.

[4] Marco Cuturi and Mathieu Blondel.
Soft-dtw: a differentiable loss function for time-series.
In *International Conference on Machine Learning (ICML)*, pages 894–903, 2017.

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

[6] Anthony Florita, Bri-Mathias Hodge, and Kirsten Orwig.
Identifying wind and solar ramping events.
In *2013 IEEE Green Technologies Conference (GreenTech)*, pages 147–152. IEEE, 2013.

[7] Laura Frías-Paredes, Fermín Mallor, Martín Gastón-Romeo, and Teresa León.
Assessing energy forecasting inaccuracy by simultaneously considering temporal and absolute errors.
Energy Conversion and Management, 142:533–546, 2017.

[8] Damien Garreau, Sylvain Arlot, et al.
Consistent change-point detection with kernels.
Electronic Journal of Statistics, 12(2):4440–4486, 2018.

[9] Rob Hyndman, Anne B Koehler, J Keith Ord, and Ralph D Snyder.
Forecasting with exponential smoothing: the state space approach.
Springer Science & Business Media, 2008.

References II

[10] Shuang Li, Yao Xie, Hanjun Dai, and Le Song.
M-statistic for kernel change-point detection.
In *Advances in Neural Information Processing Systems (NIPS)*, pages 3366–3374, 2015.

[11] 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.

[12] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio.
N-BEATS: Neural basis expansion analysis for interpretable time series forecasting.
International Conference on Learning Representations (ICLR), 2020.

[13] Quentin Paletta, Guillaume Arbod, and Joan Lasenby.
Benchmarking of deep learning irradiance forecasting models from sky images–an in-depth analysis.
Solar Energy, 2021.

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

[15] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski.
Deep state space models for time series forecasting.
In *Advances in neural information processing systems (NeurIPS)*, pages 7785–7794, 2018.

[16] Charles Truong, Laurent Oudre, and Nicolas Vayatis.
Supervised kernel change point detection with partial annotations.
In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3147–3151. IEEE, 2019.

[17] Loïc Vallance, Bruno Charbonnier, Nicolas Paul, Stéphanie Dubost, and Philippe Blanc.
Towards a standardized procedure to assess solar forecast accuracy: A new ramp and time alignment metric.
Solar Energy, 150:408–422, 2017.

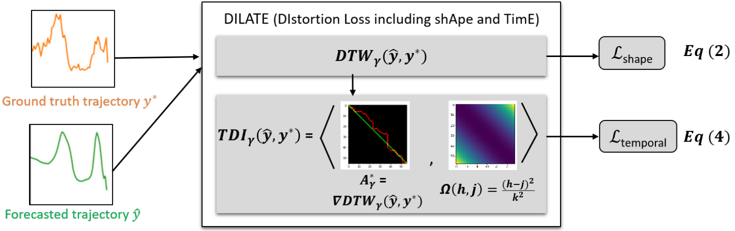
References III

[18] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon.
Temporal regularized matrix factorization for high-dimensional time series prediction.
In *Advances in neural information processing systems (NIPS)*, pages 847–855, 2016.

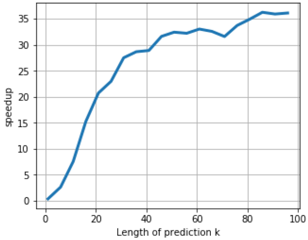
[19] Rose Yu, Stephan Zheng, and Yan Liu.
Learning chaotic dynamics using tensor recurrent neural networks.
In *ICML Workshop on Deep Structured Prediction*, volume 17, 2017.

[20] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang.
Informer: Beyond efficient transformer for long sequence time-series forecasting.
AAAI, 2021.

Training deep forecasting models with DILATE



- Direct computation of $\mathcal{L}_{shape}$ and $\mathcal{L}_{temporal}$ intractable ($|\mathcal{A}_{k,k}| = O(exp(k^2))$)
- Solution: dynamic programming $\Rightarrow$ custom forward/backward implementation



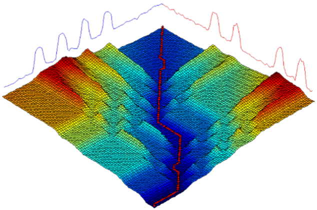
Variants of DILATE

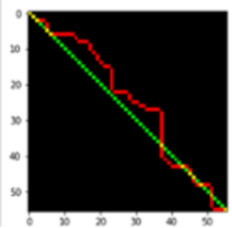
- DILATE-t: "tangled" variant of DILATE

DILATE	$\min_{\gamma}^{A} \langle \mathbf{A}, \mathbf{\Delta} \rangle + \langle A^*, \mathbf{\Omega} \rangle$
DILATE-t	$\min_{\gamma}^{A} \langle \mathbf{A}, \mathbf{\Delta} + \mathbf{\Omega} \rangle$

- DILATE-t: penalization matrix $\mathbf{\Omega}$ inside the minimization of DTW
  - Shape and temporal term mixed during minimization
- DILATE-t subsumes well-known temporally-constrained DTW methods:

Sakoe-Chiba hard band constraint	$\Omega(h,j) = +\infty$ if $\lvert h-j \rvert > T$, 0 otherwise
Weighted DTW	$\Omega(h,j) = f(\lvert i-j \rvert)$, f increasing function





DILATE experiments

		Fully connected network (MLP)			Recurrent neural network (Seq2Seq)		
Dataset	Eval	MSE	DTW_γ	DILATE (ours)	MSE	DTW_γ	DILATE (ours)
Synth	MSE	**1.65** ± **0.14**	4.82 ± 0.40	**1.67**± **0.184**	**1.10** ± **0.17**	2.31 ± 0.45	**1.21** ± **0.13**
	DTW	38.6 ± 1.28	**27.3** ± **1.37**	32.1 ± 5.33	**24.6** ± **1.20**	**22.7** ± **3.55**	**23.1** ± **2.44**
	TDI	15.3 ± 1.39	26.9 ± 4.16	**13.8** ± **0.712**	17.2 ± 1.22	20.0 ± 3.72	**14.8** ± **1.29**
ECG	MSE	**31.5** ± **1.39**	70.9 ± 37.2	37.2 ± 3.59	**21.2** ± **2.24**	75.1 ± 6.30	30.3 ± 4.10
	DTW	19.5 ± 0.159	18.4 ± 0.749	**17.7** ± **0.427**	17.8 ± 1.62	17.1 ± 0.650	**16.1** ± **0.156**
	TDI	**7.58** ± **0.192**	38.9 ± 8.76	**7.21** ± **0.886**	8.27 ± 1.03)	27.2 ± 11.1	**6.59** ± **0.786**
Traffic	MSE	**0.620** ± **0.010**	2.52 ± 0.230	1.93 ± 0.080	**0.890** ± **0.11**	2.22 ± 0.26	**1.00** ± **0.260**
	DTW	24.6 ± 0.180	**23.4** ± **5.40**	**23.1** ± **0.41**	24.6 ± 1.85	**22.6** ± **1.34**	**23.0** ± **1.62**
	TDI	**16.8** ± **0.799**	27.4 ± 5.01	**16.7** ± **0.508**	**15.4** ± **2.25**	22.3 ± 3.66	**14.4**± **1.58**

Table: Forecasting results with MSE, shape & time metrics (10 runs, avg ± std). For each experiment, best method(s) (Student t-test) in bold.

Evaluation with external metrics

- Shape: **ramp score** [17]
- Time: **Hausdorff distance** between 2 sets of change points

		MSE	DTW_γ	DILATE (ours)
Synthetic	Hausdorff	2.87 ± 0.127	3.45 ± 0.318	**2.70 ± 0.166**
	Ramp score (x10)	5.80 ± 0.104	**4.27 ± 0.800**	4.99 ± 0.460
ECG5000	Hausdorff	**4.32 ± 0.505**	6.16 ± 0.854	**4.23 ± 0.414**
	Ramp score	**4.84 ± 0.240**	**4.79 ± 0.365**	**4.80 ± 0.249**
Traffic	Hausdorff	**2.16 ± 0.378**	**2.29 ± 0.329**	**2.13 ± 0.514**
	Ramp score (x10)	6.29 ± 0.319	**5.78 ± 0.404**	**5.93 ± 0.235**

Table: Forecasting results of Seq2Seq evaluated with Hausdorff and Ramp Score, averaged over 10 runs (mean ± standard deviation). For each experiment, best method(s) (Student t-test) in bold.

Comparison to tangled variants of DILATE

Eval loss		DILATE (ours)	DILATE^t-Weighted	DILATE^t-Band Constraint
Euclidian	MSE (x100)	**1.21** ± **0.130**	1.36 ± 0.107	1.83 ± 0.163
Shape	DTW (x100)	**23.1** ± **2.44**	**20.5** ± **2.49**	**21.6** ± **1.74**
	Ramp	**4.99** ± **0.460**	**5.56** ± **0.87**	**5.23** ±**0.439**
Time	TDI (x10)	**14.8** ± **1.29**	17.8 ± 1.72	19.6 ± 1.72
	Hausdorff	**2.70** ± **0.166**	**2.85** ± **0.210**	3.30 ± 0.273

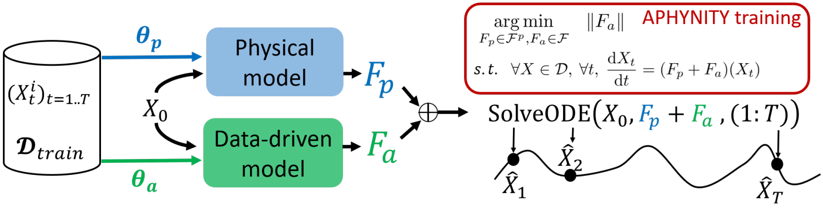
Table: Comparison to the tangled variants of DILATE for the Seq2Seq model on the Synthetic dataset, averaged over 10 runs (mean ± standard deviation).

Baselines:

- N-Beats [12]
- Informer [20]

Dataset	Model	MSE	DTW	Ramp	TDI	Hausdorff	DILATE
Synthetic	N-Beats [16] MSE	**13.6 ± 0.5**	24.9 ± 0.6	5.9 ± 0.1	**13.8 ± 1.1**	**2.8 ± 0.1**	**19.3 ± 0.5**
	N-Beats [16] DILATE	**13.3 ± 0.7**	**23.4 ± 0.8**	**4.8 ± 0.4**	**14.4 ± 1.3**	**2.7 ± 0.5**	**18.9 ± 0.8**
	Informer [17] MSE	**10.4 ± 0.3**	20.1 ± 1.1	4.3 ± 0.3	13.1 ± 0.9	**2.5 ± 0.1**	16.6 ± 0.8
	Informer [17] DILATE	11.8 ± 0.7	**18.5 ± 1.2**	**2.4 ± 0.3**	**11.6 ± 0.9**	**2.4 ± 0.9**	**15.1 ± 0.7**
Electricity	N-Beats [16] MSE	**24.8 ± 0.4**	**15.6 ± 0.2**	**13.3 ± 0.3**	4.6 ± 0.1	**2.6 ± 0.3**	**13.4 ± 0.2**
	N-Beats [16] DILATE	25.8 ± 0.9	**15.5 ± 0.2**	**13.3 ± 0.3**	**4.4 ± 0.2**	3.1 ± 0.5	**13.2 ± 0.2**
	Informer [17] MSE	**38.1 ± 2.1**	18.9 ± 0.6	13.2 ± 0.2	6.5 ± 0.3	2.1 ± 0.2	16.4 ± 0.5
	Informer [17] DILATE	**37.8 ± 0.8**	**18.5 ± 0.3**	**12.9 ± 0.2**	**5.7 ± 0.2**	**1.9 ± 0.1**	**15.9 ± 0.3**
ETTH1 (96)	N-Beats [16] MSE	32.5 ± 1.4	3.9 ± 0.2	13.3 ± 2.0	21.6 ± 4.3	**5.7 ± 0.7**	7.4 ± 1.0
	N-Beats [16] DILATE	**26.0 ± 2.8**	**2.9 ± 0.1**	**4.6 ± 0.6**	**11.4 ± 1.7**	**6.4 ± 1.0**	**4.6 ± 0.4**
	Informer [17] MSE	**28.2 ± 2.6**	4.3 ± 0.3	5.8 ± 0.1	21.6 ± 3.3	**6.6 ± 1.9**	7.8 ± 0.9
	Informer [17] DILATE	32.5 ± 3.8	**3.2 ± 0.3**	**4.5 ± 0.3**	**19.1 ± 1.9**	**6.4 ± 1.0**	**6.4 ± 0.6**

APHYNITY optimization



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

$$\mathcal{L}_{\lambda_j}(\theta_p, \theta_a) = \|F_a^{\theta_a}\| + \lambda_j \cdot \mathcal{L}_{traj}(\theta_p, \theta_a) \quad (4)$$

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

- $\theta = (\theta_p, \theta_a)$, 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**
      $\theta_{j+1} = \theta_j - \tau_1 \nabla [\lambda_j \mathcal{L}_{traj}(\theta_j) + \|F_a\|]$
  $\lambda_{j+1} = \lambda_j + \tau_2 \mathcal{L}_{traj}(\theta_{j+1})$