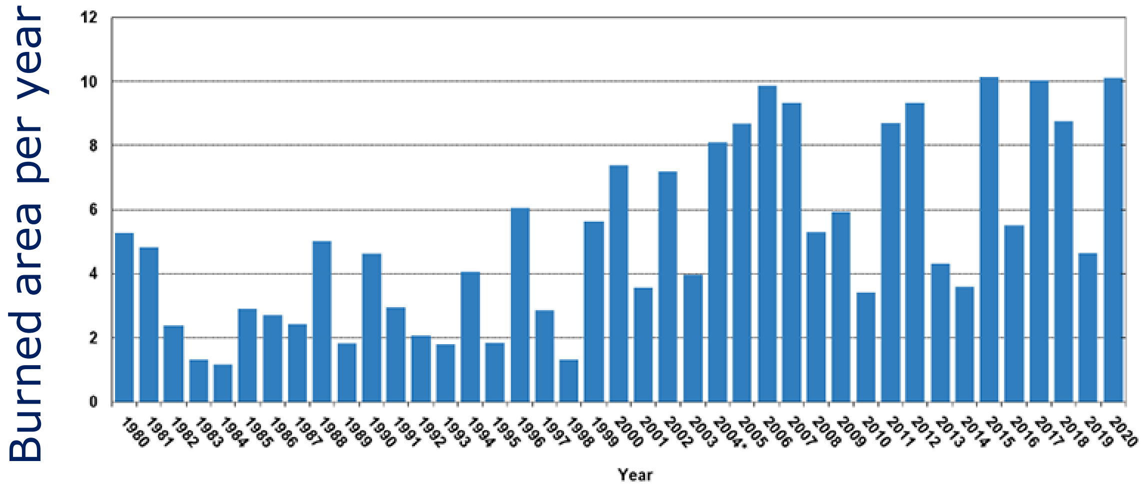
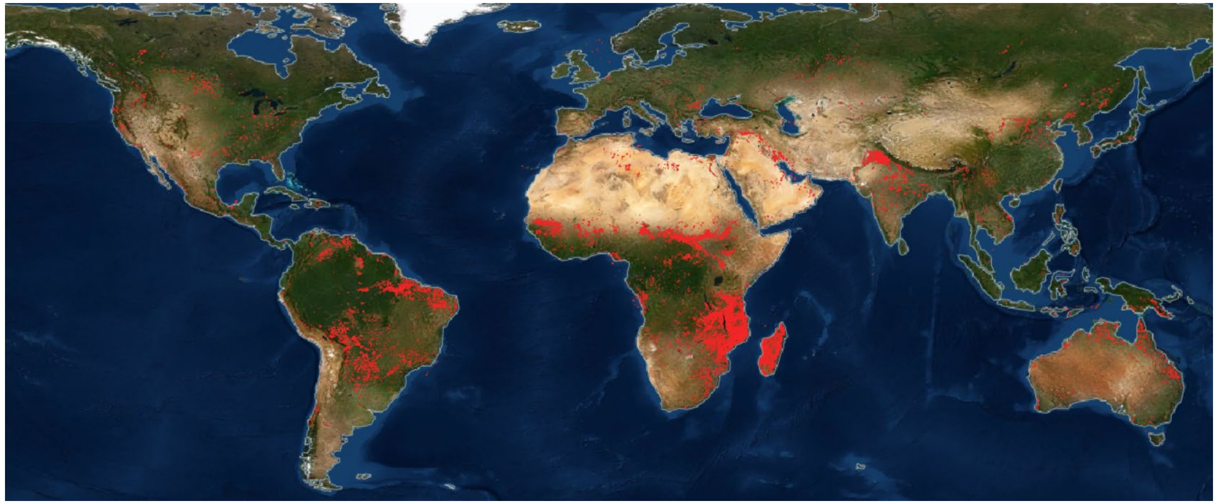
Data-Driven Surrogate Model with Latent Data assimilation for Wildfire Forecasting

Sibo Cheng, Colin Prentice, Yike Guo and Rossella Arcucci
Imperial College London

Motivation



The recent wildfires in California in 2018 cost more than $27 billion capital loss



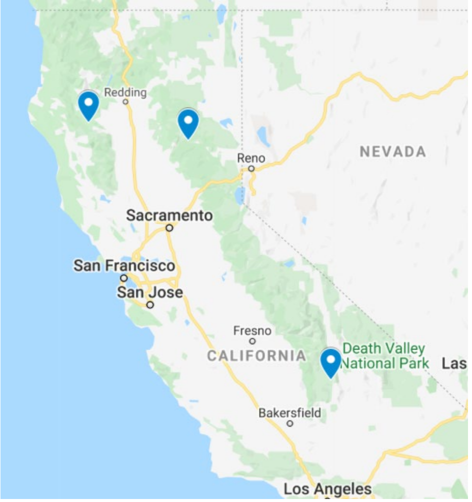
https://firms.modaps.eosdis.nasa.gov/map

04/11/2021

Study areas and observations

Daily satellite (MODIS) observation

Images every 1-2 days at 1km resolution



Idea:

- Learning from simulation data
- Using satellite observations to validate/assimilate

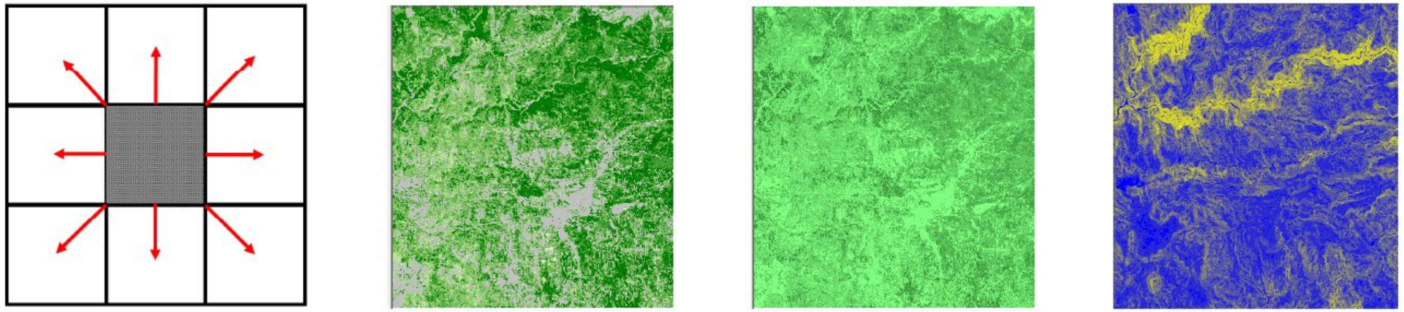
Buck fire, 2017

Fire	latitude		longitude		area
	North	South	West	East	
Bear 2020	39.8567	39.7780	-121.1615	-121.0171	$\approx 108\text{km}^2$
Buck 2017	40.2558	40.1707	-123.0791	-122.9734	$\approx 83\text{km}^2$
Pier 2017	36.1909	36.0543	$-118.798\,698$	$-118.616\,145$	$\approx 244\text{km}^2$

Table 1: study areas of the three wildfires

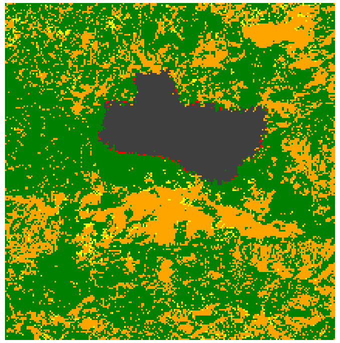
data learning

Cellular Automata (CA) simulator



(a) (b) (c) (d)

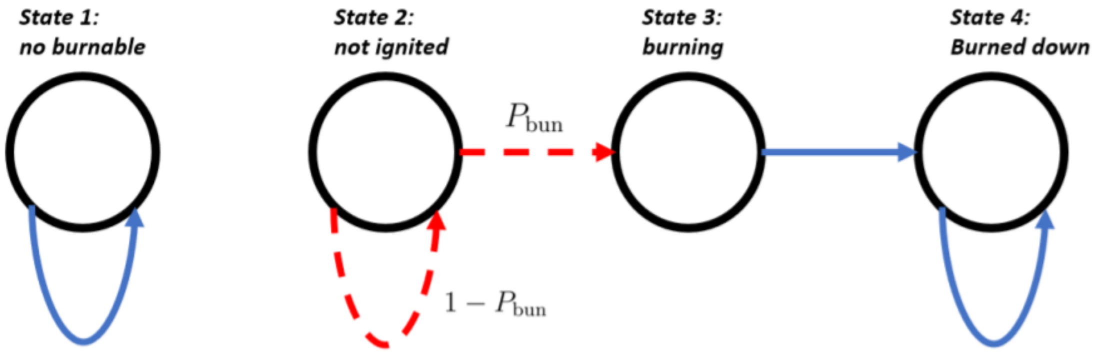
CA (Buck fire)



Satellite observations

Stochastic simulation

$$P_{\text{bun}} = p_h(1 + p_{\text{veg}})(1 + p_{\text{den}})p_s$$



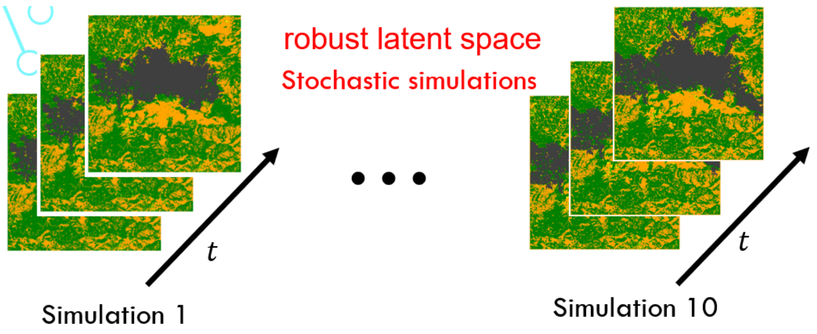
Other fire simulators/algorithms

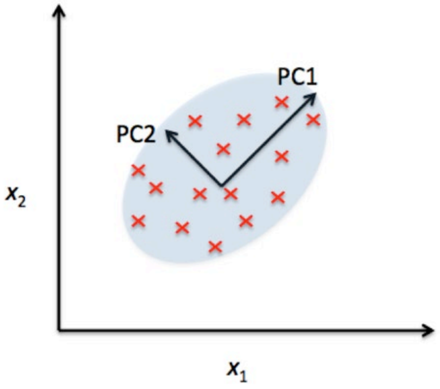
- Rothermel equation
- Flammap
- SPARK
- CA

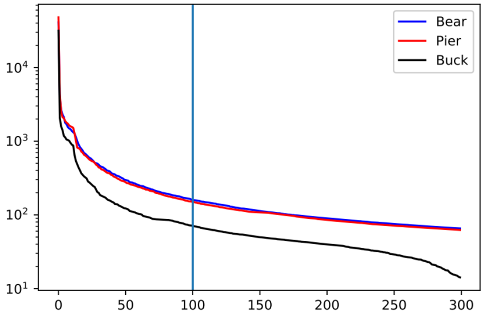
It is time consuming to simulate large fires

Reduced order modelling

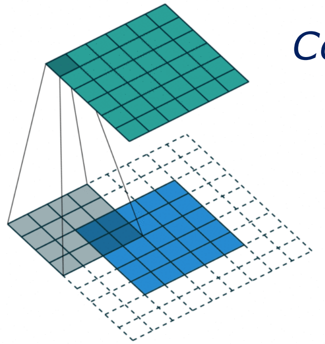
Principle component analysis (PCA)

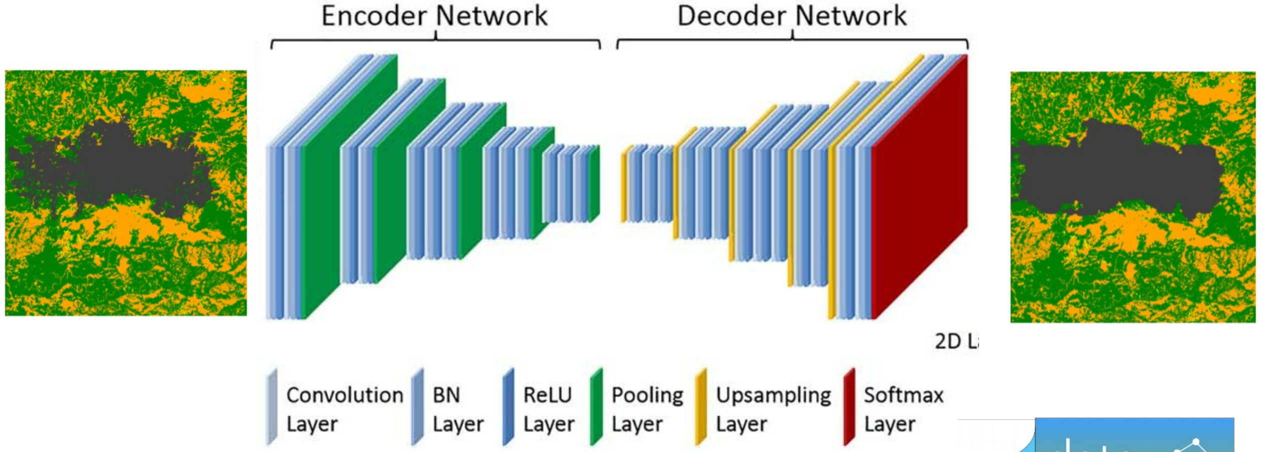






Variational auto-encoder

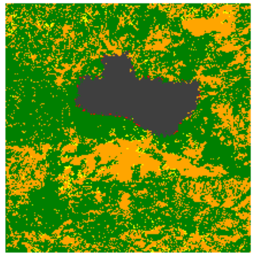




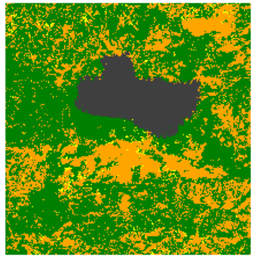
CAE vs. POD (reconstructions)

Training set: 120 CA simulations
Satellite obs: Daily observation (MODIS)

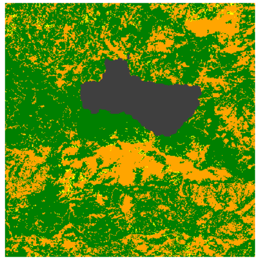
CA

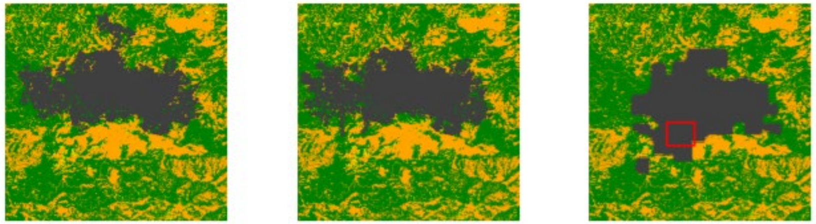


POD

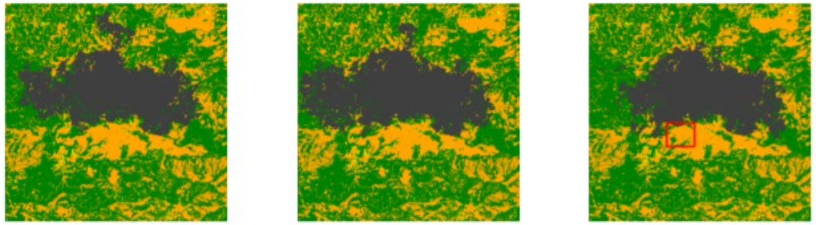


CAE

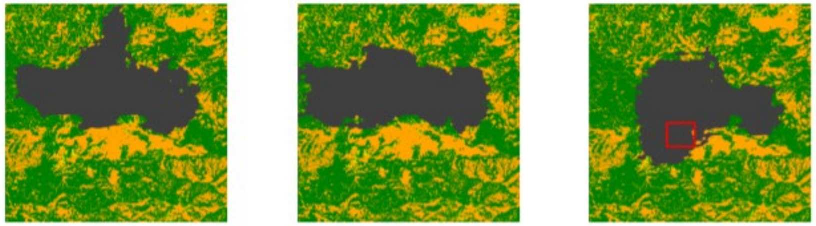




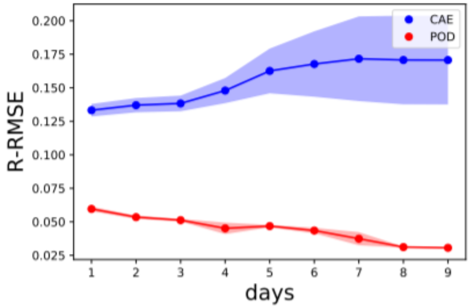
(a) training: full (b) test: full (c) obs: full

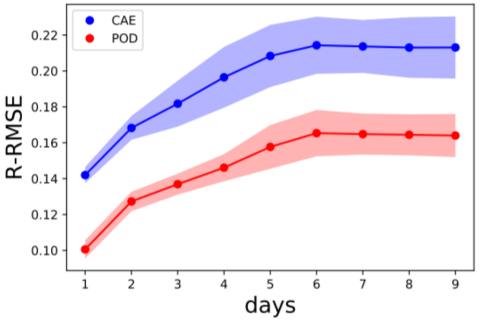


(d) training: POD (e) test: POD (f) obs: POD

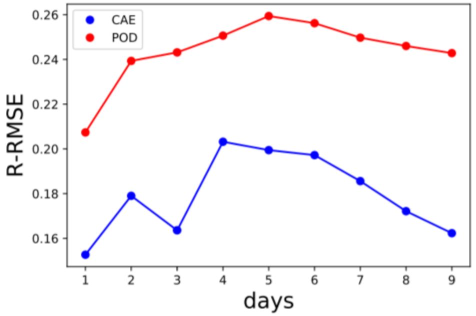


(g) training: AE (h) test: AE (i) obs: AE





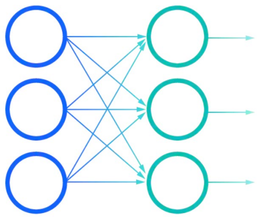
(b) testing



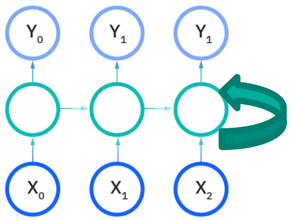
(c) observations

Latent RNN

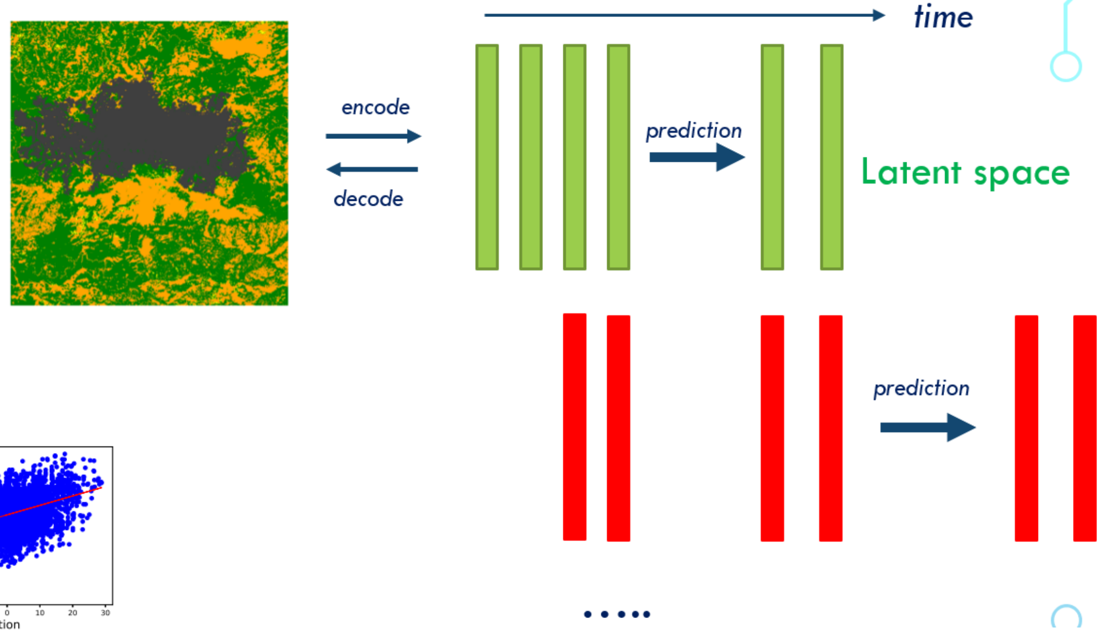
DNN



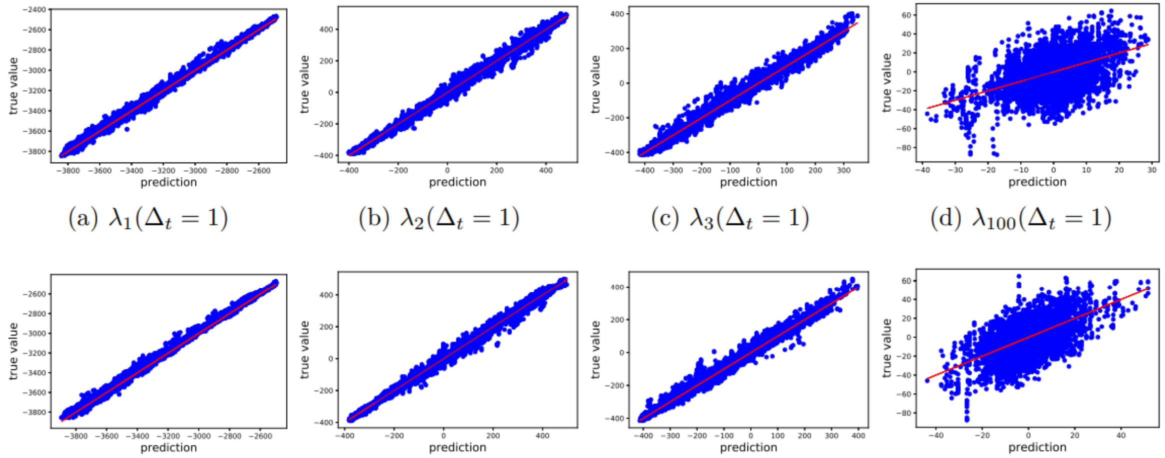
RNN



RNN in the latent space (many-to-many LSTM)



Prediction in the latent space



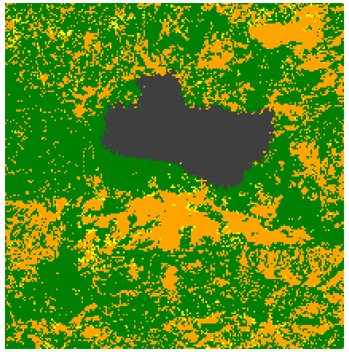
(a) $\lambda_1(\Delta_t = 1)$ (b) $\lambda_2(\Delta_t = 1)$ (c) $\lambda_3(\Delta_t = 1)$ (d) $\lambda_{100}(\Delta_t = 1)$

(e) $\lambda_1(\Delta_t = 10)$ (f) $\lambda_2(\Delta_t = 10)$ (g) $\lambda_3(\Delta_t = 10)$ (h) $\lambda_{100}(\Delta_t = 10)$

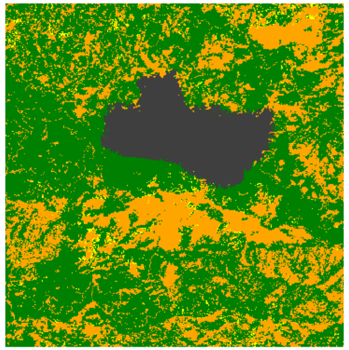
data learning

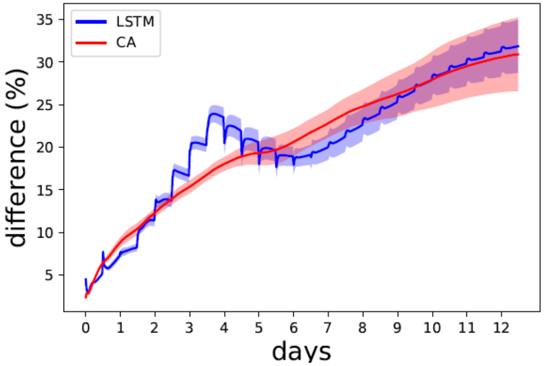
Prediction error

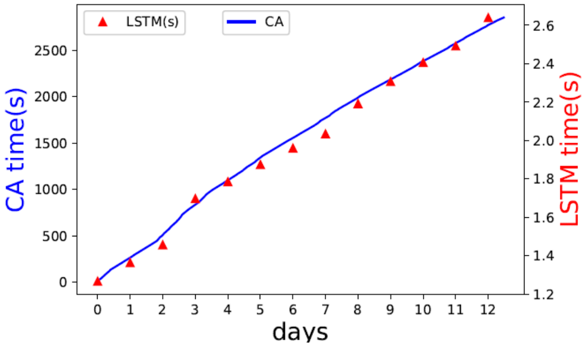
reconstruction



prediction

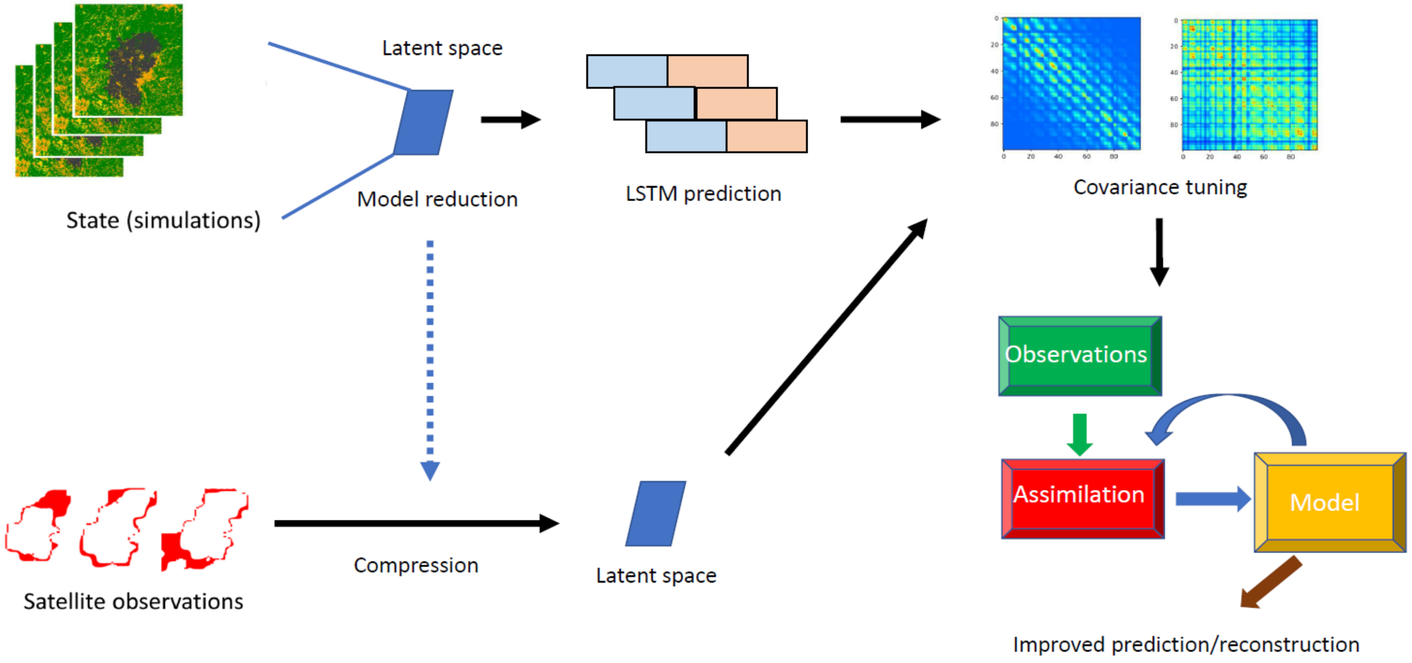






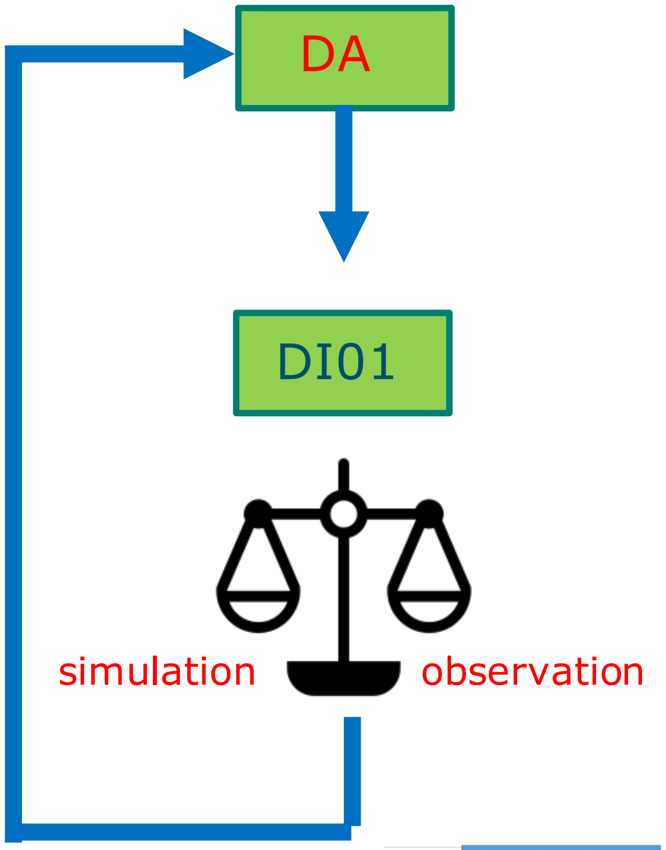
How to correct the predictions with real-time observations?

Latent Assimilation: principle

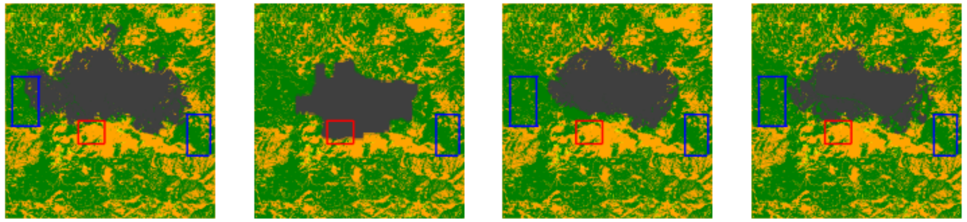


DI01 covariance tuning

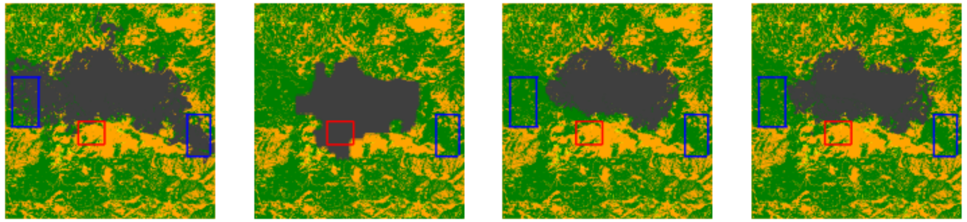
Desroziers & Ivanov, 2001



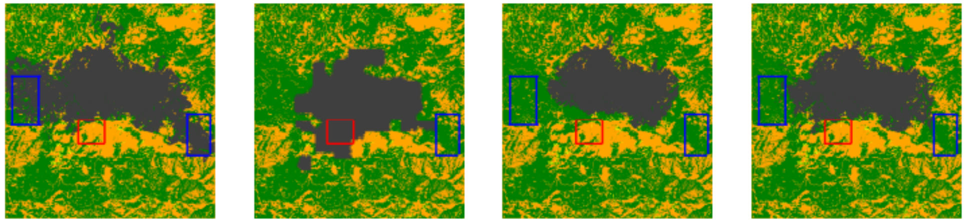
Latent Assimilation: results



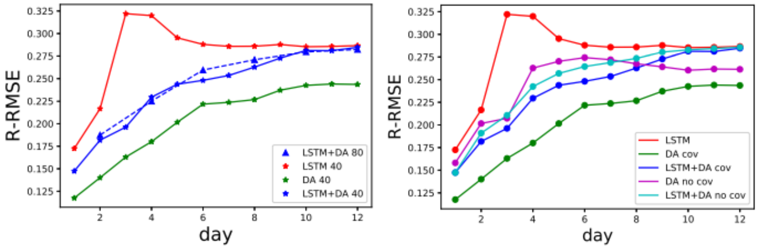
(e) LSTM(day 2) (f) obs(day 2) (g) LSTM+DA(day 2) (h) DA(day 2)



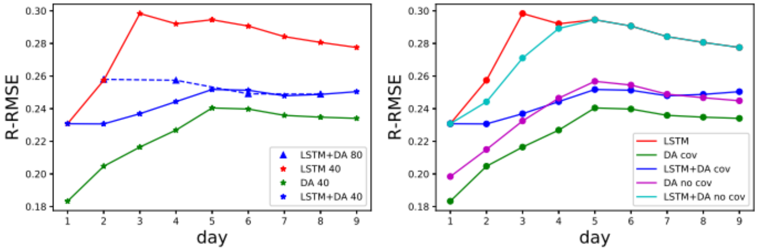
(i) LSTM(day 3) (j) obs(day 3) (k) LSTM+DA(day 3) (l) DA(day 3)



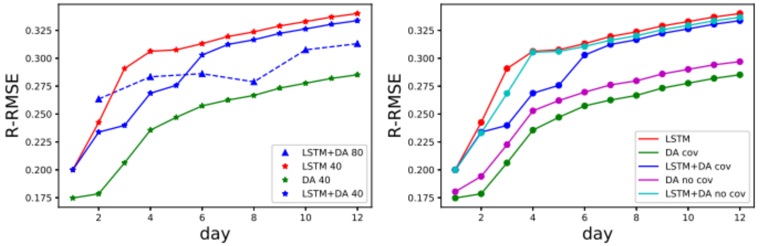
(m) LSTM(day 4) (n) obs(day 4) (o) LSTM+DA(day 4) (p) DA(day 4)



(a) Bear 2020: latent LSTM and DA (b) Bear 2020: effect of the covariance tuning



(c) Buck 2017: latent LSTM and DA (d) Buck 2017: effect of the covariance tuning



Fire	simulation	LSTM	DA	DI01	Flammap
Bear 2020	$3.85s$	$3.1e^{-3}s$	$5.8e^{-3}s$	$8.7e^{-2}s$	$10 \sim 30s$
Buck 2017	$2.96s$	$5.26e^{-3}s$	$4.26e^{-3}s$	$2.12e^{-2}s$	$5 \sim 20s$
Pier 2017	$8.28s$	$6.28e^{-3}s$	$5.36e^{-3}s$	$2.68e^{-2}s$	$10 \sim 30s$

Table 5: Averaged computational time for one time-step using different approaches

Conclusion and future work

Conclusion

- The ROM- and RNN-based surrogate model is very efficient
- Latent Assimilation is computationally cheap and can be performed near real-time

Future works

- Physics-informed machine learning
- Variational latent assimilation
- More general machine learning modelling for fire spread prediction