Different ideas for Application of ML/AI in Data Assimilation

Miodrag Rancic
Lynker at NOAA/NCEP/NWS/EMC

Introduction

- Rapid development and application of Artificial Intelligence and Machine Learning (AI/ML) in literally all fields of human endeavor, and consequently in meteorology and atmospheric sciences in general, was mostly caused by two factors:
  - Flood of observations and accumulation of various other information on atmosphere (reanalysis, models outputs, etc.)
  - Development and application of GPU (Graphics Processing Units) which enabled extremely fast processing of large amount of parallel data, making them far more efficient than CPUs for deep learning and neural network training.
- ML emulators of weather forecasting models have already achieved significant respect due to their efficiency, but also ability to achieve very good results.
- Examples are
  - GraphCast, FourCasttNe, Pangy-Weather for global scales
  - StormCast and ML LAM for mesoscale limited area models
  - Aardvark Weather – end-to-end ML forecasting system

ML in Data Assimilation and development of a ML RTMA

- While ML has been applied to different aspects of Data Assimilation, its application as an emulator of a full DA system is still in the infancy stage.
- Real Time Mesoscale Analysis (RTMA) under development at EMC and Global System Laboratories (GSL) is a state-of-the-art project that provides a near-real time analysis for the contagious territory of the United States, Alaska, Quam and Hawaii at a very large horizontal spatial resolution of 2.5 km, targeting future extensions up to 1 km, being issued at similarly frequent time intervals of 15 min. A major prerequisite for success of this project is a large computational efficiency.
- Therefore, we started looking for different methods to develop a ML version of RTMA.
- This talk will summarize some of the major candidates for this task, and along way share some lessons learnt on general application of ML in DA.

General direction

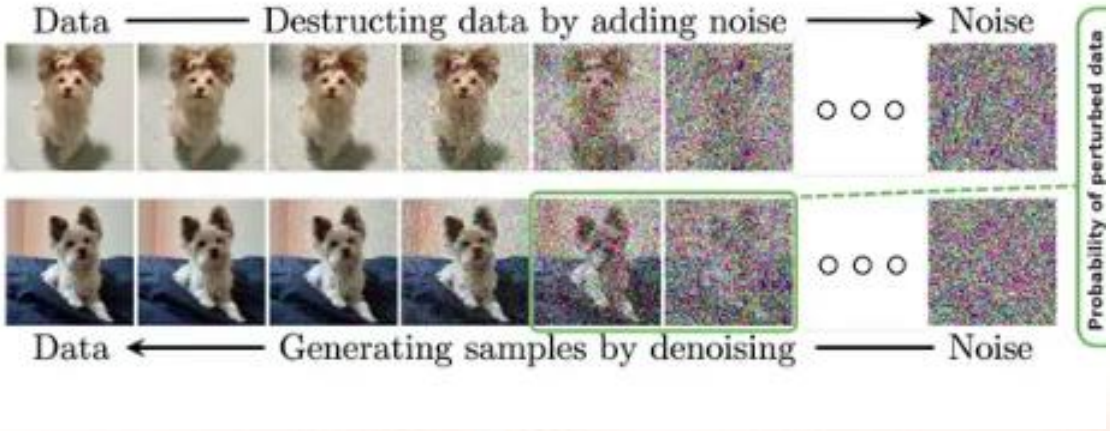
- Success of machine learning models critically depend on the quality of data used as a reference (labels) for training.
- Therefore, development of ML DA system does not assume abolishment of standard DA.
- Quite the opposite! We assume that standard DA (in the case of RTMA, called URMA), will continue to be improved through application of the most advanced classical methods, such as 4D VAR hybridized with ensemble Kalman filter, regardless of computational coast.
- Such DA system will be used off-line for re-training of ML RTMA (among other things).
- At the same time, ML RTMA will be run in the online mode with supposedly a substantially reduced execution time.

Considered methods (so far)

- Denoising diffusion (Ho et al., 2020)
- Adaptation of end-to-end data driven approach, based on application of transformers.
- A 4D VAR with ML versions of TL and AD
- Application of PINNs (Physically Informed NN) to provide data at points of a regular grid
- Two examples of a `home made' convolutional U-net developed for experimentation and learning.

Generative diffusion method

- Diffusion models belong to the group of generative AI models.
- The idea is that by gradually adding Gaussian noise to the image in the forward diffusion process, we came to a nosy pattern, and then by going in the opposite direction and using conditioning we lead the denoising toward new imagers.



Some References:

1. What are Diffusion Models? — https://lilianweng.github.io/posts/2021-07-11-diffusion-models
2. How diffusion models work: the math from scratch — https://theaisummer.com/diffusion-models
3. Introduction to Diffusion Models for Machine Learning — https://www.assemblyai.com/blog/diffusion-models-for-machine-learning-introduction
4. Diffusion Models Made Easy — https://towardsdatascience.com/diffusion-models-made-easy-8414298ce4da
5. Diffusion Models: A Comprehensive Survey of Methods and Applications — https://arxiv.org/pdf/2209.00796.pdf

Denoising diffusion (Ho et al., 2020)

- The denoising approach (DiffDA) for DA is described in Huang et al. (2024).
- It is based on the process of denoising diffusion (Ho et al., 2020) applied to assimilation of atmospheric variables, using predicted states and sparse observations – which perfectly suits objectives of transitioning of RTMA to an AI/ML version.
- Original DiffDA is applied to the global coverage and is based on pretrained global ML model GraphCast as the core of their diffusion model.
- During denoising stage, they apply conditioning methods developed for stable diffusion (Rombach, et al. 2022), combined with soft masking and interpolation of observations, guides the denoising process toward the desired assimilation state.

Pros and cons

- This method successfully leverages a pre-trained ML based model by exploiting Its learned representation of atmospheric dynamics, making the assimilation process more efficient and accurate.
- In comparison with traditional methods, while its training is reported to be rather expensive, once fDiffDA is trained and when running in the inference mode, its efficiency is quite superior.
- The conditioning mechanism allows for flexible integration of diverse and sparse observations and allows using new observations that are added at inference time without retraining.
- Similarly, the method is robust to missing data, since it learns from patterns and can infer information.
- Being incorporated alongside with application of a ML forecasting model, this approach opens a pathway toward a fully data driven, ML-only pipeline.

Adaptation of DiffDA for a regional domain and in RTMA

- In the case of ML RTMA, we adopted a ML-LAM (Oskarsson et al., 2024) as a ML forecasting model to be part of DiffDA approach.
- Diffusion-LAM (Larsson et al., 2205) is also available and could be used in this project.
- Different mechanisms are available to control relative weights given to different observations during the denoising process.
- For example, the attention mechanism, applied along with the U-net, which is traditionally used in diffusion models, can inherently learn to weight the importance of different parts of the input (equivalent to considering observational error).
- The denoising method can automatically bring the probabilistic component to a DA system.

Adaptation of end-to-end data driven approach

- This ML method avoids using background fields entirely and instead produces a future atmospheric state at a regular grid starting directly from raw observations (e.g., Zhao et al., 2024, Allen et al., 2025).
- This method is based on application of vision transformer (Dosovitskiy et al., 2021), which generally includes a sequence of
  - encoding (where impact of observations are in one or another way projected to the whole domain)
  - the processing, and
  - decoding, generally based on a convolutional architecture.
- In application to DA the method needs to be modified since we are not interested in forecast.
- In the case of RTMA, one concern with application of this approach is that a relative sparsity of observations and insufficient satellite imagery may lead to an inadequate description of atmospheric dynamics.

A 4DVAR with ML versions of TL and AD models

- Tangent linear (TL) and adjoint (AD) of ML weather emulators (e.g., Tian, et al., 2023) can be applied in the context of a full 4DVAR.
- There is a concern with the physical reality of ML TL and AD (Tian et al., 2024).
- This issue might be addressed through a hybridization of ML emulator with application of physical constraints added as a weak constraint to loss function during training.
- Here, the idea is to regularize the loss function by pushing the produced variables during training to obey underlying atmospheric dynamics.
- Generally, such an attempt for hybridization is expected to stabilize and help data driven model to better deal with situations in which it was not trained.

Example of inclusion of physics constraint in a ML emulator

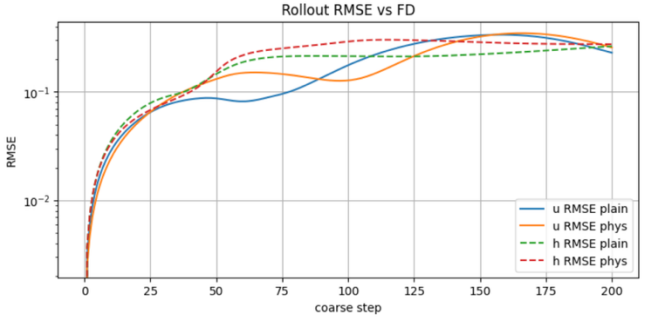
- One-dimensional shallow water model

$$\partial_t u + u\partial_x u + g\partial_x h = \nu\partial_{xx} u$$

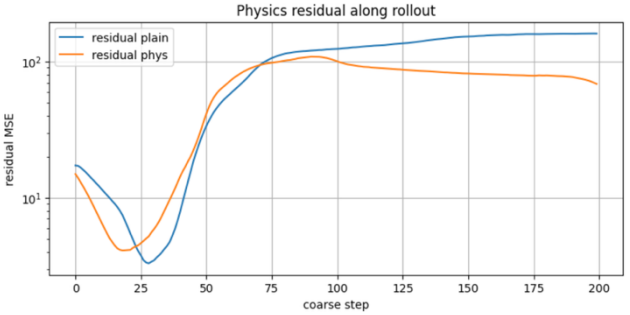
$$\partial_t h + \partial_x (uh) = \nu\partial_{xx} h$$

- U-net ML model was run in an autoregressive mode, with “large” time-steps.

$$\mathcal{L}_{\text{phys\_total}} = \mathcal{L}_{\text{data}} + \lambda\,\mathcal{L}_{\text{phys\_resid}}$$



Evolution of RMSE during inference stage. (Lower is better).



Physics residual MSE measures violation of PDE during inference. (Lower is better).

Application of PINNs to provide data at points of a regular grid

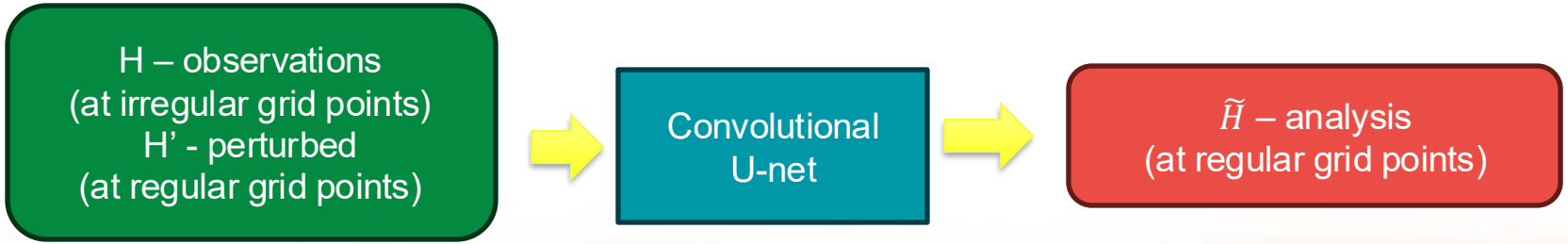
- Physically Informed Neural Networks (PINNs) represents a method for solving partial differential equations by imbedding physical laws in the NN architecture.
  - Raissi et al., (2019) – original paper
  - Cuomo et al., (2022) – a comprehensive review
- Soto et al., (2024) present method to derive gridded weather conditions from the sparse observations, using PINNs.
- They use observations from surface stations to present weather at points of a regular mesh.
- This method could be very interesting as a specific ML 4D DA approach, capable of picking up available data from the whole assimilation window.
- It might be applied in the first stage to add a temporal dimension to 2D RTMA.

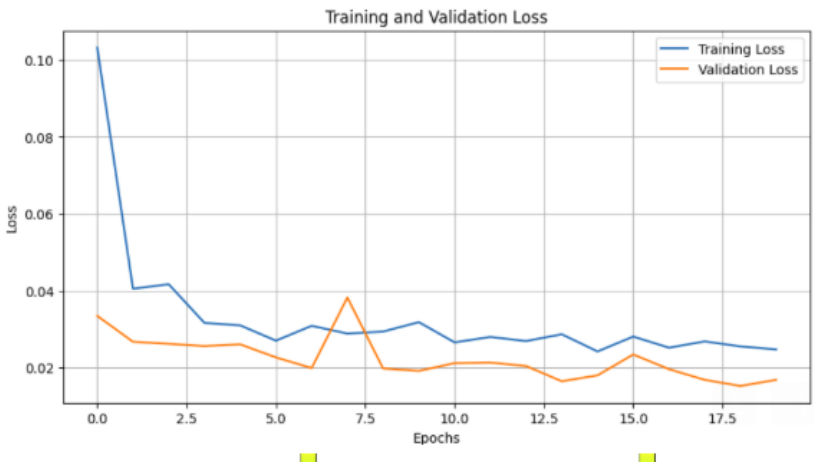
Additional comments

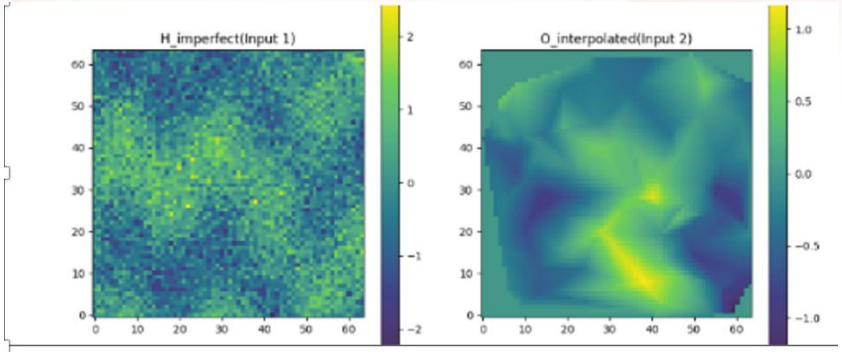
- Once the first version of the ML DA is finished, one can always additionally improve its performance by including sources of high-quality fields of some of critically important components of reference fields, such as cloud ceiling and visibility, which could be incorporated into re-training of the ML RTMA at later time.
- The technique which could be used to this end is referred to Transfer Machine Learning (Zhuang et al., 2020).

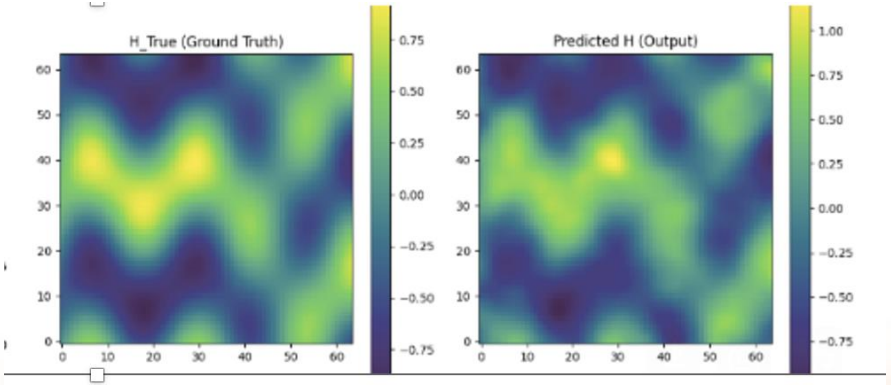
Examples of first toy models of ML DA – Exercise 1

- Define a 2D field (H).
- Select a randomly distributed set of points that will play role of ”observations”.
- Encode ”observations” to the same grid at which background is defined (O).
- Perturb slightly original field to get an ”imperfect background” (B).
- Apply a convolutional U-net with two channels at input (O and B) and one at the output $\widetilde{H}$.
- Train U-net using original field (H) as ground truth.



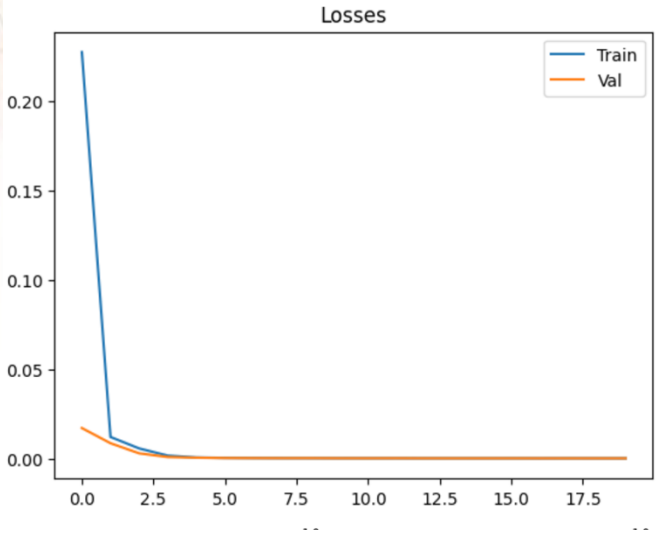

Training and Validation Loss
Training Loss
Validation Loss
Loss
Epochs
0.10
0.08
0.06
0.04
0.02
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

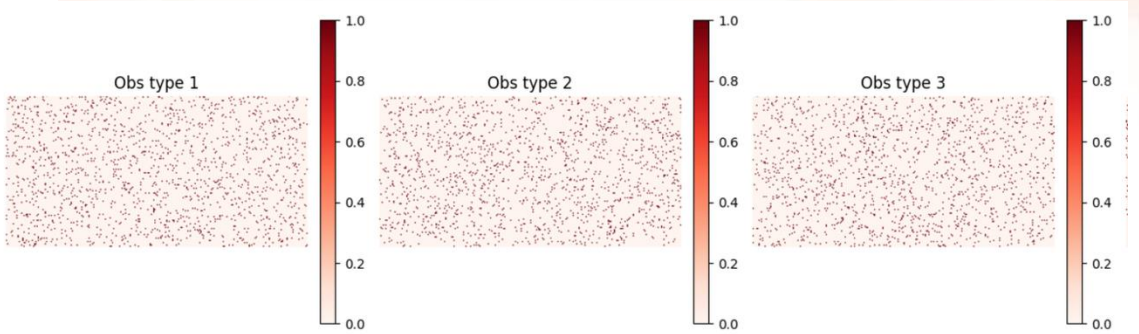

H_imperfect(Input 1)
O_interpolated(Input 2)

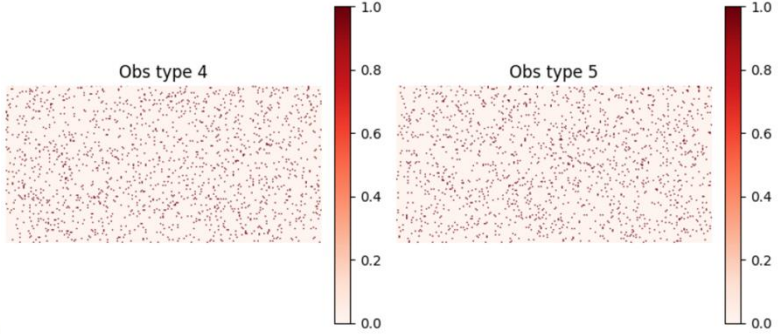

H_True (Ground Truth)
Predicted H (Output)

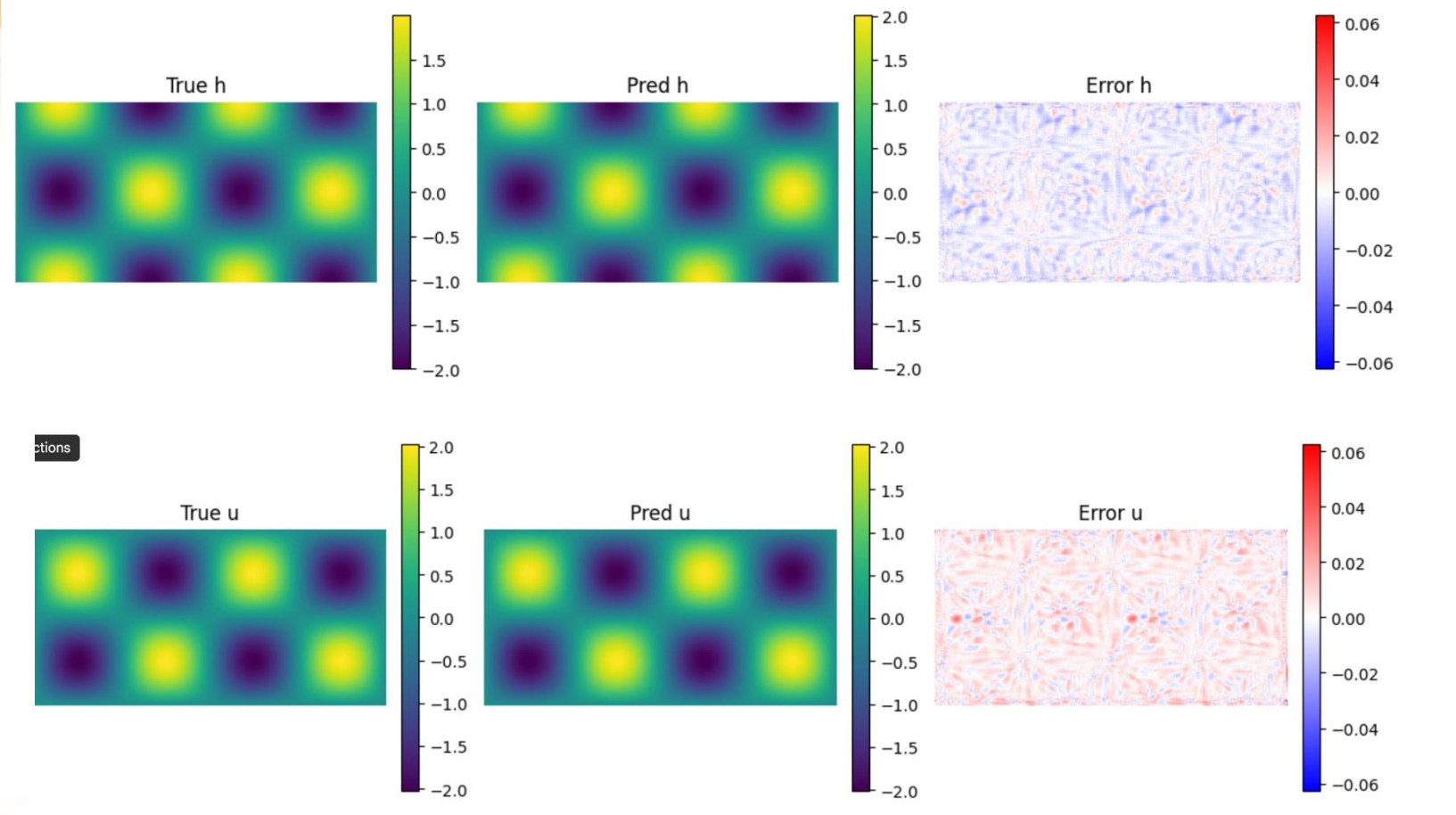
Examples of first toy models of ML DA – Exercise 2

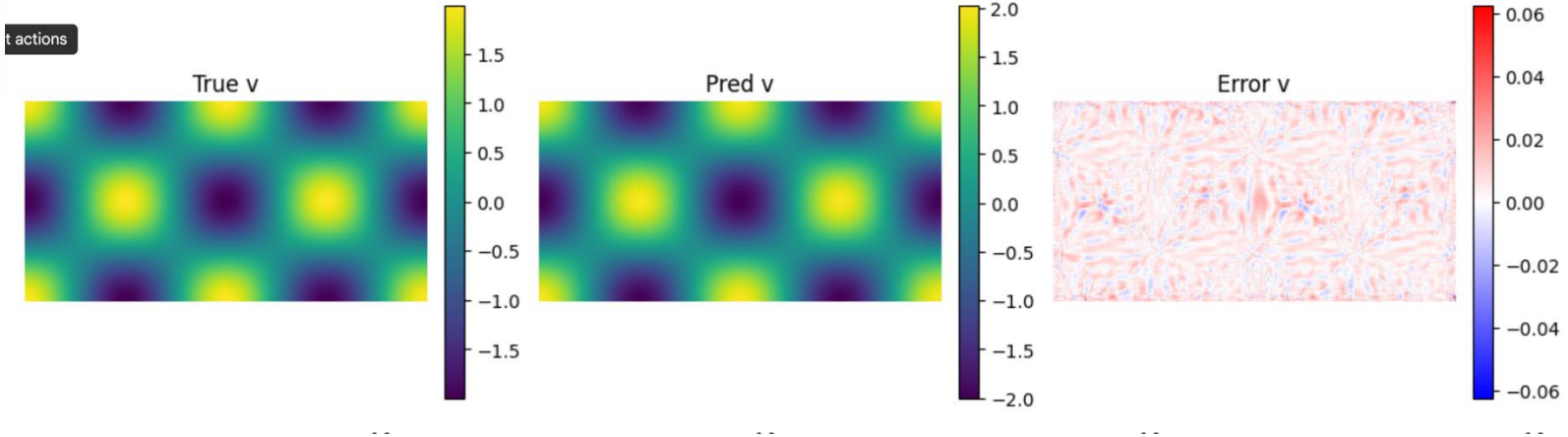
- Assume that there is no background
- Use full shallow water model with three variables (h, u, v)
- Assume that we have observations of all three fields defined at random points in the domain.
- In addition, assume that we have two types of observations, which could be expressed as some functions of base variables (o1, o2)
- We are looking for analysis defined at a set of regular grid points, using again a convolutional U-net, this time with 5 channels at input and 3 at output:


Losses
Train
Val
0.20
0.15
0.10
0.05
0.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


Obs type 1
Obs type 2
Obs type 3
1.0
0.8
0.6
0.4
0.2
0.0


Obs type 4
Obs type 5
1.0
0.8
0.6
0.4
0.2
0.0


True h
Pred h
Error h
True u
Pred u
Error u
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06


t actions
True v
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
Pred v
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
Error v
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06

Conclusion

- ML DA assimilation is making huge strides, and it is very interesting to see what comes next.
- This was in no way a complete list of ongoing work in this arena, and the main motivation was to encourage exploration of this interesting new field.

Thank you!