Geoscientific Model Development Discussions

Geosci. Model Dev. Discuss., referee comment RC1
https://doi.org/10.5194/gmd-2021-71-RC1, 2021

Referee comment on gmd-2021-71

Anonymous Referee #1

Referee comment on "Towards physics-inspired data-driven weather forecasting: integrating data assimilation with a deep spatial-transformer-based U-NET in a case study with ERA5" by Ashesh Chattopadhyay et al., Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2021-71-RC1, 2021

This manuscript explores improvements in the rapidly advancing field of data-driven weather prediction (DDWP). Broadly, DDWP seeks to train empirical weather-prediction models based on deep learning architectures, such as convolutional neural networks, that have proved very successful in fields such as image processing. This work fits within the WeatherBench forecasting challenge, which aims to forecast the global 500hPa geopotential height field, given the same field at an earlier time.

One of the leading approaches for DDWP is to use a convolutional U-NET architecture in which the first ("encoding") half projects the higher-resolution geopotential height field onto one or more lower-resolution "latent spaces" or "encoding spaces" and the second ("decoding") half of the U-NET upsamples the results, via many convolutional layers, to the original space. Broadly, the convolutional blocks are learning how to project geopotential height features forward in time, with the different levels of the U-NET allowing different scales to be projected using different convolutional blocks. The first advance is, at the lowest-level encoding space of the U-NET, to add an "equivariance-preserving spatial transformer"; the resulting network is known as U-STN and improves forecast quality over the U-NET. The spatial transformer appears to permit additional capabilities for rotation, scaling and translation of the encoded geopotential height features within the empirical model, which are helpful for improving the forecast performance. The addition of the spatial transformer is justified as providing additional capabilities to preserve equivariance to important symmetries in the fluid dynamics of the atmosphere and therefore to provide a more physics-aware neural network. I would like to see more justification for this interpretation, and more precision in its discussion (see below).

The second advance in the manuscript is to couple a data assimilation algorithm to the DDWP model. Currently the Weather Bench framework provides high-quality gridded initial conditions from which to run DDWP forecasts, therefore missing a major step in the broader challenge of weather forecasting, which is to create those gridded initial conditions by assimilating the diverse and sparse (non-gridded) weather observations. To

explore this side of the problem, noise is added to the gridded initial conditions, which are then assimilated every 24h using an ensemble data assimilation algorithm. An interesting aspect is to use the low-cost DDWP to create much larger ensembles (around 4000 members) than are possible in typical NWP (around 50 - 100 members). This allows a novel ensemble DA algorithm to be used (one that can be coded in a few lines of python), apparently without the problems of covariance localisation that are required with smaller ensembles. A second application of DA is also presented, where it is used to merge forecasts from DDWP models with different integration lengths.

This is novel and interesting work, which may have substantial impact on the development of DDWP, and hence it is worthy of eventual publication. However, there are a few major issues to consider beforehand, including the previously mentioned issues around the physical interpretation.

Main issues

1) The idea of equivariance is introduced precisely in Wang et al. (2020), for example, as applying to a function f(x) given a symmetry group g. The function is equivariant to g if the result of applying any of the symmetries (or transformations) from the group is the same whether applied to the functions inputs or outputs: f(g x) = g f(x). The same paper lists the symmetries of the Navier-Stokes equations as space and time translation, uniform motion, reflect/rotation and scaling. By contrast, the current manuscript is in places vague about what it means by equivariance, and it does not anywhere show whether it is preserved in the models presented. The presentation and analysis of the results relating to equivariance needs to be improved:

(i) Through the manuscript there are statements referring to the U-STN as "the equivariance-preserving DDWP introduced here" (line 118). However, the baseline U-NET is also likely to be equivariance-preserving, at least to translation and reflection. The improved U-STN may add equivariance to a certain set of symmetries (the authors suggest reflection, rotation and scaling). The point being that both the U-NET and the U-STN are likely equivariance-preserving to some degree, but neither of them in a complete way to all possible symmetries. In terms of analysis, it would be important to more precisely specify or confirm which symmetries are preserved, or to acknowledge if the exact set of symmetries preserved is unknown. In terms of presentation, to describe the U-STN as equivariance-preserving and to imply the U-NET is not could be misleading, and the title might also be changed to better reflect this.

(ii) This work adds an affine transformation and an interpolation (manuscript equations 1 and 2) in the latent space of the encoder; this is referred to as a "spatial transformer" and described as creating a new coordinate system, which is then passed to the decoding part of the U-NET. On line 138 - 139 it is said "The spatial transformer module ensures that the latent space that is encoded is equivariance-preserving". First, given the definition of equivariance, it is hard to see how a latent space could be equivariance-preserving. Rather, it would be the relevant function, i.e. the spatial transformer, that is equivariance-preserving. In any case, this assertion needs to be properly backed up. As a concrete

example, to be equivariance-preserving to rotations, it would beed to be shown that all rotations of features in the encoding space (the input to the spatial transformer) would provide identical results to those performed in the transformed space acted on by the decoder (the output of the spatial transformer).

(iii) An alternative explanation for the success of the U-STN would be to think of the spatial transformer as being able to learn a transformation that is helpful to propagating the encoding-space version of the geopotential height field forward in time. The spatial transformer is described by a single 2x3 transformation matrix, T(theta), with 6 trainable parameters (manuscript equation 2), followed by an interpolation This can only learn to perform one transformation, and for example, it might have learnt a particular combination of rotation and translation helpful to propagating the encoding space equivalents of Rossby waves forward in time. To better understand what is going on from a physical point of view, it would be really helpful if the authors could present the 6 parameters of T(theta) and try to interpret their effect in these terms: what does the learned transformation do (e.g rotation, scaling, translation?), does it make physical sense?

2) The level of methodological detail in the manuscript is not fully sufficient to allow replication of the results or to communicate the approach at a sufficient level of detail. The neural networks being used are not fully described in the manuscript. A better example would be Weyn et al. (2020) who have shown how it is possible to properly document a complex network structure within a paper, such as by providing a table describing the layers, tensor sizes, etc.. It would also be helpful to have more details on the technical implementation such as the use of Python, Keras and Tensorflow, for example.

3) Some source code is provided on Xenodo, and it helped me a lot in understanding the work. However, it still left a lot unclear, and I believe it may only be a sample from all the code used by the authors while performing their work. For example, the training details appear to have been placed within a Jupyter notebook (Unet_STN.ipynb), but it is not fully clear whether this applies to all three examples in the manuscript, and to both the U-NET and the U-STN, and to the three different training time ranges (1,3 or 12 h), which is unlikely. The definition of the U-STN network in the Jupiter notebook is very different from the ones in the EnKF examples, which is confusing - see attached file "u_stn_diff.txt". It is not clear whether the U-net definition is provided at all. I would have expected a standardised definition of the networks in a separate file that could be used by all different configurations. Generally, the code package could be made more helpful to other people by better documentation and/or comments, better code structure and standardisation, and by the provision of some or all of the relevant data files - in particular the network weights of the U-NET and U-STN.

Minor issues

1) Line 24: "…promising results with fully data-driven weather prediction (DDWP) models that are trained on variables representing the large-scale circulation obtained from numerical models or reanalysis products (Scher, 2018; Weyn et al., 2019, 2020;

Chattopadhyay et al., 2020d, a; Rasp et al., 2020; Arcomano et al., 2020; Chantry et al., 2021; Grönquist et al., 2021; Watson-Parris, 2021; Scher and Messori, 2021)". Not all of the citations here are presenting the DDWP of the large-scale circulation - for example, Watson-Parris (2021) and Chantry et al. (2021) are opinion pieces and Grönquist et al. (2021) concerns postprocessing. This is a helpful bibliography and I am not suggesting the removal of any of the citations. Rather it might be worth giving a few more words to categorise these works more precisely. Further, this list is missing a key reference in Rasp and Thuerey (2020), which is discussed by the authors just afterwards.

2) Line 30: "… DDWP models may not suffer from some of the biases of physics-based, operational numerical weather prediction (NWP) models …". It seems unnecessarily restrictive to mention only bias here; the aim is to reduce model uncertainty in general.

3) Line 40: "… to equip these DDWP models with data assimilation (DA) …". As written, the role of data assimilation is left uncertain. Although DA is introduced more fully later in the introduction, it could still be helpful to give slightly more clarity here, for example "to run these DDWP models within a data assimilation framework to provide the initial conditions for the forecasts".

4) Line 68 gives the first mention of the U-NET architecture in the paper; a citation or two might be handy, and/or a pointer to the parts of the paper that describe what it is.

5) Line 74: "DA algorithm that corrects the trajectory of the atmospheric states every 6 h with observations from remote sensing and in-situ measurements" - every 6h is overly restricitve, ERA5 for example is produced on a 12h cycle.

6) Line 117 "The baseline DDWP model used here is a U-NET similar to the one used in Weyn et al. (2020)" - as in main point 2, I would have found it helpful to have more description of the baseline U-NET, and it would be nice to know more precisely what is different compared to Weyn et al.

7) Line 118 mentions the deep spatial transformer in the method section for the first time; a citation to the original source would be helpful here, and also in section 3.1.2 where it is described in more detail.

8) In the bibliography, the citation to Esteves et al. (2018) is mostly in lowercase.

9) Line 152 - 153: "All codes for these networks (as well as DA) have been made publicly available on GitHub (see the Code Availability statement)." The codes are provided on Zenodo (not GitHub) but as described in main point 3, they do not appear to be complete.

10) Line 164 - please give a few words of explanation on the meaning of "unscented"

11) Line 184: in the DA algorithm, the singular vector decomposition of the analysis error covariance matrix is used to generate perturbations to create a new ensemble. However, in this work the ensemble is not propagated forward in time hour-by-hour, but is generated using the analysis error valid at "t" to represent the forecast error at "t+23dt", which is strictly incorrect. The forecast error at 23h is going to be much larger than the analysis error at 0h, therefore the ensemble created in the current work is most likely an underestimate of the spread of the background error. This needs to be discussed in the manuscript.

12) Equations 6 and 8 have identical right hand sides, but are labelled as different things (P_a and P_ab respectively). So something must be missing from the RHS to explain why they are different, or else P_a and P_ab are the same.

10) I found Figure 2 and 7 slightly confusing. The positioning of the states Z(t), Z(t+ dT) and so on below the U-STN1 blocks is confusing if the x-axis represents time; the small blue arrows are not helpful (suggesting that the states are external data coming into the process) and not consistently applied either. It could be more helpful to more clearly show the relation of the U-STN1 blocks to their inputs and outputs.

11) The forecast verification in Figure 3 is based on 30 random initial conditions (line 249). Seeing as it is so cheap to run the DDWP models, why not provide the results based on the full 2018 test period, to obtain more statistical significance? It is also odd to see the strong variability in skill, particularly in the U-STN12, from one verification time to the next. This might suggest that the verification is not as statistically significant as suggested by the standard deviation range provided. Verification of NWP forecasts is usually much smoother as a function of forecast range. Even for DDWP forecasts such as shown in Weyn et al. (2020, their figs 4 and 5) this usually seems to be the case.

12) Another point on the comparison of U-STN12 to U-NET12, it would be really helpful to establish the quality of the U-NET12 baseline - how competitive is it with other DDWP models?

13) Line 261-262: "The reason behind the further improvement of the performance after DA is the de-noising capability of neural networks (Xie et al., 2012)" - this seems overly confident given that it has not been demonstrated in the manuscript: "A likely reason behind the further improvement …" would be a fairer description.

14) Section 4.3 gives an example of the use of DA to merge forecasts of different lengths.

I find this section helpful as an illustration of the skill variations obtained with cycled (autoregressive) predictions versus direct predictions. However, instead of using DA, why not just throw away the cycled U-STN1 state at t+12dt and replace it by the forecast from U-STN12? It would be good to see if the DA can actually improve on that; in other words whether the cycled U-STN1 is bringing some additional information that is worth preserving.

16) Conclusions / discussion: on the benefits of DDWP for DA algorithms, item 2 line 314: being able to generate an ensemble large enough to provide fully-sampled background error covariance matrix is a major benefit here. However, the state vector size in the current work (2048) is still quite small compared to what might be expected in a more sophisticated DDWP approach, let alone NWP, where the state vector size is approaching 10^8. It should be acknowledged and discussed that the ability to use the SPEnKF algorithm, and to dispense with localisation, is not just the speed of the model (the DDWP) but the small size of the state vector.

References (see manuscript for others)

Wang, R., Walters, R. and Yu, R., 2020. Incorporating symmetry into deep dynamics models for improved generalization. arXiv preprint arXiv:2002.03061.

Weyn, J. A., Durran, D. R., & Caruana, R. (2020). Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. Journal of Advances in Modeling Earth Systems, 12, e2020MS002109. https://doi.org/10.1029/2020MS002109