Referee 1

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.

Authors' response:

We thank the referee for their positive evaluation of our manuscript. Based on the referee's suggestion we have revised our manuscript in blue. The referee's insightful suggestions have sufficiently improved the clarity of the manuscript. Herein, we provide point-by-point responses to the referee's comments.

Referee's comment:

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 USTN 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 USTN 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.

Authors' response:

We thank the referee for raising this interesting point about symmetry groups. In the manuscript, we have explained that equivariance-preserving networks do not impose *a priori* symmetry inside the networks and rather optimizes parameters to learn the symmetry. It is also true that U-STN does not learn all symmetries. In our U-STN, we have implemented rotational, translational, and scaling transformation through six parameters in the $T(\theta)$ matrix (defined in section 3.1.2) connected to the latent space of the network. It does not impose rotational invariance, i.e., it does not enforce the output to remain invariant to rotation in the input. The spatial transformer module learns the transformation (only on the encoded latent space) such that the latent space decodes to the correct output (which may have undergone rotational, translational, or scaling transformation) during training. The U-NET, like a regular CNN, is invariant to translation. In the U-STN, we have only performed rotational, translational, and scaling transformation on the latent space of the encoder which we have further clarified in the revised manuscript between Lines 150-152. We have changed the title to remove the word equivariance-preserving (and spatial transformers) and have kept the word geometric deep learning. Equivariance is a topic of interest in the geometric deep learning community [1] and we have thus added a reference to a recent survey in geometric deep learning (Bronstein et al., 2021) in Table 1.

Referee's comment:

(*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 need 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).*

Authors' response:

We thank the referee for their insightful comment. By saying that the "latent space" is equivariance preserving, we meant to say that the nonlinear function that operates on the latent space of the U-NET captures the transformation given by $T(\theta)$ between the input of the latent space and the output of the decoder and is clarified in Line 149 in the revised manuscript. However, given the complexity of the feature space in the Z500 field, it is difficult to interpret whether the transformation in the latent space of the U-NET leads to physically meaningful transformations in the decoded output. For a simpler dataset such as the rotating MNIST, spatial transformer networks have been shown to capture meaningful rotational features in Jaderberg et al., [2]. We show that, in data-driven forecasting of weather, the spatial transformation (in the latent space) allows us to obtain a better prediction horizon as compared to the U-NET (without such a transformation). However, we agree with the referee that such interpretations of the transformation in the latent space should be pursued further in future studies. We have revised our manuscript between Lines 262-265 to reflect the difficulty in interpreting the precise effect of the transformation induced by $T(\theta)$ in complex physical flows such as the large-scale circulation.

Referee's comment:

(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?

Authors' response:

We thank the referee for their insightful comment. Firstly, we agree with the referee that $T(\theta)$ learns only one transformation between the latent space of the network and the decoded output which is a combination of rotation, translation, and scaling. It is difficult to precisely interpret how the transformation given by $T(\theta)$ in the latent space captures specific features in a complex flow field such as Z500. We have tried interpreting the parameters, θ, but because $T(\theta)$ is only applied to the latent space, it is very difficult to perform any interpretations as to which features are captured in the complex Z500 field. We agree with the referee that with further development in geometric deep learning [1], such exercises in interpretation, especially in complex physical flows should be undertaken in the future. We believe however, that we should first start with simpler atmospheric models such as quasi-geostrophic flow where such analysis towards interpretation should be performed. We are currently working towards such interpretation through a hierarchy of simpler atmospheric/ocean models.

Referee's comment:

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.

Authors' response:

We thank the referee for their helpful suggestion. Following the referee's suggestion, we have added Table 2 in the revised manuscript where we have documented the detailed difference between U-NET and U-STN architecture and the framework in which they have been implemented.

Referee's comment:

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.

Authors' response:

We thank the referee for going through our code in such details and providing helpful suggestions to improve the readability of our code. We have worked on organizing our source code more carefully and have now provided the networks' weights and biases. We have uploaded the same on Zenodo and updated our Github.

Referee's comment:

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.

Authors' response:

We thank the referee for this helpful suggestion. We have revised Lines 25-31 to incorporate this suggestion.

Referee's comment:

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.

Authors' response:

We thank the referee for this helpful suggestion. We have revised Line 34 to incorporate this suggestion and referred to the "biases of physics-based, operational numerical weather prediction (NWP) models" as model errors in general.

Referee’s comment:

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”.

Authors’ response:

We thank the referee for this helpful suggestion. We have revised Lines 46-47 in the revised manuscript to incorporate this suggestion.

Referee’s comment:

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.

Authors’ response:

We thank the referee for this helpful suggestion. We have now added a reference to the original U-NET paper in Line 75 in the revised manuscript.

Referee’s comment:

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.

Authors’ response:

We thank the referee for pointing this out. We have revised the manuscript (Line 81) to say that 6 h is an example time interval at which DA is performed.

Referee's comment:

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.

Authors' response:

We thank the referee for this helpful suggestion. We have revised the manuscript between Lines 179-184 to briefly talk about the difference between the architectures of the two U-NETs. However, we want to emphasize that we are not presenting a benchmark DDWP model that competes with the DDWP model in Weyn at al., [4]. In this manuscript, we are providing a proof-of-concept for a specific spatial transformation module in the latent space that may improve the performance of any DDWP architecture, and we have considered U-NET as an example.

Referee's comment:

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.

Authors' response:

We have added the reference to the original paper in Line 142.

Referee's comment:

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

Authors' response:

Thank you. We have fixed the reference.

Referee's comment:

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.*

Authors' response:

The codes in the Github repository are complete but we have only uploaded the U-STN for one Δt where that variable can be changed to incorporate any other Δt. We have now organized our source code so that it is readable and has more clarity. We thank the referee for taking the time to go through our codes and providing us with useful suggestions.

Referee's comment:

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

Authors' response:

We have now added a reference to unscented transformations in Line 187.

Referee's comment:

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.

Authors' response:

We thank the referee for pointing this out. Carrying the ensembles for 24 h is computationally expensive especially since the ensemble size is very large (4096). We agree that the ensemble spread is an underestimate of the background error in this work and have highlighted that in the revised manuscript in Lines 212-213. However, we have performed experiments by propagating the ensembles as well and have not found a significant difference in performance. This has also been reported in the revised manuscript in Lines 211-212.

Referee's comment:

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.

Authors' response:

Thank you for pointing out this typo. We have fixed the equations in the revised manuscript.

Referee's comment:

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.

Authors' response:

Thank you for this helpful suggestion. We have slightly changed the schematic to adjust the time stamps of the U-STN blocks.

Referee's comment:

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.

Authors' response:

We have conducted a Kolmogorov-Smirnov test between the difference of the mean ACC of U-STN12 and U-NET12. The difference is **not** statistically significant between 12 h and 36 h, it **is** statistically significant between 36 h and 240 h.

Referee's comment:

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?

Authors' response:

We have revised the manuscript between Lines 275-280 to compare between other baselines such as CNN and linear regression. However, we would like to emphasize that, in this paper, we do not intend to present the best DDWP model but rather a proof-of-concept on the advantage of using an equivariance-preserving module in the latent space and a framework to integrate DA with DDWP. Moreover, herein we present Table. 1 which compares the quality of performance of U-STN12, U-NET12 and the CNN and linear regression model in WeatherBench [5].

Table 1. A comparison between U-STN12 and U-NET12 presented in this manuscript and the linear regression and CNN models from WeatherBench [5]. Here, we have used the RMSE values of direct linear regression and CNN prediction at day 3 and day 5. Here "direct" refers to the models trained to predict Z500 directly at 3 and 5 days instead of iteratively predicting every hour.

Models	RMSE (m^2s^{-2}) at 3 days	RMSE (m^2s^{-2}) at 5 days
Linear regression (direct) from WeatherBench	693	783
CNN (direct) from WeatherBench	626	757
U-STN12 (our model)	**294**	**490**
U-NET12 (our baseline model)	**310**	**517**

Referee's comment:

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.

Authors' response:

Thank you. We have revised the manuscript in Lines 298-300 based on the referee's suggestion.

Referee's comment:

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.

Authors' response:

We agree with referee's point. The directly predicted state at $t + 12\Delta t$ would probably be skillful as well. In fact, Liu et al., [3] had used interpolation instead of DA to obtained skillful autoregressive prediction with a multi-step framework. Further, not using DA would reduce the computational cost of the framework as well. However, the skill of the prediction by using the predicted state at $t + 12\Delta t$ from U-STN12 also depends on the value of initial noise as well. A fair comparison would require conducting further systematic experiments to see how the effect of noise affects recycling the state at $t + 12\Delta t$ in comparison to performing DA with it. In this paper, we show that performing DA with it is simply one way to recycle the obtained state and there may well be other methods such as interpolation (as shown in Liu et al., [3]) or simply using the state itself for further predictions that may yield skillfull predictions.

Referee's comment:

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.

Authors' response:

Thank you. We have revised the text accordingly between Lines 356-357.

References:

1. Bronstein, M., Bruna, J., Cohen, T., and Velickovic, P., Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges, arXiv preprint arXiv:2104.13478,2021.

2. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks, in: Advances in Neural Information Processing Systems, pp. 2017–2025, 2015.

3. Liu, Y., Kutz, J. N., and Brunton, S. L., Hierarchical Deep Learning of Multiscale Differential Equation Time-Steppers, arXiv preprintarXiv:2008.09768, 2020.

4. Weyn, J.A., Durran, D.R., and Caruanna, R., Improving Data-Driven Global Weather Prediction Using Deep Convolutional Neural Networks on a Cubed Sphere, of Advances in Modeling Earth Systems, 12(9):e2020MS002109, 2020.

5. Rasp, S., Dueben, P. D., Scher, S., Weyn, J. A., Mouatadid, S., and Thuerey, N.: WeatherBench: A Benchmark Data Set for Data-Driven Weather Forecasting, Journal of Advances in Modeling Earth Systems, 12, e2020MS002 203, 2020.