Referee 1.

Referee comment: Most of my comments have been addressed by a significant number of changes through the manuscript, and in the improved set of code archived on Zenodo. However, there is still need for improved presentation of the relevant neural networks, in order to make sure that the work can be correctly interpreted and reproduced. I put these as "major": although they are not extensive, there is one potentially important question over the fundamental methodological description.

Authors' response: We thank the referee for their positive assessment of our manuscript. Herein, we provide a point-by-point response to the referee's comments. All changes in the manuscript have been marked in blue. We also thank the referee for their helpful suggestions that have greatly improved the clarity of the manuscript.

Referee comment: The addition of Table 1 is a big step forward in documenting the relevant neural networks used in this study. However, comparing it to the code archived on Zenodo brings a number of questions.
- It would be useful to document the 2x2 resolution of the max pooling layers. This can be inferred from the text, but it would be useful to make that explicit.
- It would be useful to document input tensor shape/size for each layer in the table - this is most useful for layers 9-12 where it has not been clearly documented in the text.
- To properly understand the U-net architecture it is necessary to document the pass-throughs, e.g. the process of concatenating the output of one of the earlier higher-resolution layers with the output of an upsampling layer. For example, based on the function unet_baseline in Unet_noSTN_lead1.py, the upsampling layer 15 (as numbered in Table 1) is concatenated with the output of layer 5. This is illustrated in Figure 1, but it is helpful to have it in Table 1 as well.

Authors' response: We thank the referee for their insightful comments. We have now documented the 2×2 pooling layers in Table 2. We have also added a column to explicitly mention that output tensor shapes in Table 2. We have also explicitly mentioned the concatenation layers and which layer it has been concatenated with in Table 2. All these changes have been marked with blue.

Referee comment: Much of the text in section 3 seems to describe a single latent space at 8x16 resolution, but the U-NET also has one at 16x32 resolution.

Authors' response: The latent space for both U-STN and U-NET is 8×16. Two pooling layers would take the original sized input from 32×64 to 8×16. However, a latent space size of 16×32 would have little to no impact on the performance of both U-NET and U-STN.

Referee comment: Based on comparing the function Unet_noSTN_lead1.py and Unet_STN_lead1.ipynb, layers 9-12 in table 1 are only present in the Unet-STN: a note "Only for STN" needs to be added as for the layers 13 and 14.

Authors' response: We thank the referee for pointing this out. The note has been added in Table 2.

Referee comment: I may have misread the code, but the code does not seem to agree with the description of applying the spatial transformer to the lowest-resolution (8x16) latent space. The application of the spatial transformer is given in Unet_STN_lead1.ipynb by
x = BilinearInterpolation(sampling_size)([inputs, locnet]) In other words, the inputs are "locnet", which is the 6 elements of the affine transformation, generated by layers 9-12 (as described in Table 1) and "inputs" which is actually the input to the very first convolutional layer, in other words the geopotential height at the 32x64 resolution. Therefore the spatial transformer seems to be applied to the full-resolution data, and not in the latent space.

Authors' response: The indices for bilinear interpolation has been obtained from the original full-sized input but the transformation has been applied to the latent space as done in the original paper by Jaderberg et al., [1].

Referee comment: Figure 1 seems to show the spatial transformer being applied to the latent space at 16x32 resolution, not the one at 8x16 resolution as described in the text, or the original 32x64 space, as I have inferred from the code.

Authors' response: The two pooling layers reduce the latent space size to 8×16 from 32×64. The transformation is applied to the latent space after it goes through the dense layers as shown in Figure 1. Not all dense layers have been shown in the figure for compactness but reported in Table 2.

Referee comment: A couple of other points:

- My request for Figure 3 to be based on a larger sample of initial conditions than 30 was not addressed in the authors' response; there seems no reason not to include as many initial conditions as possible, in order to improve the comprehensiveness of the testing, even if the authors are confident that their statistical significance levels are well estimated.

Authors' response: We thank the referee for raising this important point of uncertainty quantification of the free prediction's ACC with both U-STN and U-NET. Due to limitations in computational resources, we were not able to perform predictions with more than 30 initial conditions. However, we have systematically performed predictions with 10 and 15 initial conditions and repeated the statistical test. We have found the significance testing robust across 10, 15, and 30 initial conditions. This shows that the error bounds are robust as well in Figure 3.

Referee comment: - Line 296: "stable beyond 10 days". In Figure 5 there is a visible trend in the standard deviation (the shaded area) across the 30 initial conditions, not yet discussed in the text. This suggests that the data assimilation is not fully stable even within the 10 days.

Authors' response: We thank the referee for asking this question about the stability of the SPEnKF+U-STN1 framework. As can be seen in Figure 5, for both levels of noise, both the RMSE

(R) curves does not visibly shoot up (down) monotonically with increasing DA cycles. At every DA cycle the R curves go up and RMSE curves go down to roughly the same value at each DA cycle. The uncertainty in these curves is derived from the spread in initial conditions and not the spread of the ensembles of the analysis state. We have (not shown for brevity in the text) also investigated the background covariance matrix and it shows localized correlation structure indicating that the SPEnKF filter is not diverging up to 10 days.

References:

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

Referee 2

Thank you for thoroughly responding to the reviewers' comments: I believe the manuscript is significantly clearer and more accessible as a result. Now that the manuscript is easier to read, it is apparent that U-STN may not be equivariance-preserving (1.1) and hence physically constrained (1.2), which requires major revisions (1) as equivariance preservation is a key point of the manuscript. Making this manuscript as scientifically rigorous as possible is even more important now that its preprint is already cited four times. Minor comments are listed in Section 2; I would recommend being particularly cautious as some of the minor revisions that were discussed by the authors in the response did not make it to the "tracked changes" version that we received from GMD. Even if U-STN were not equivariance-preserving, the framework presented by the authors still improves the accuracy of U-Net and would be useful to gain insight into the meteorological prediction problem at hand (1.3), and I still recommend this manuscript for eventual publication in GMD once the network's equivariance properties are clarified.

Authors' response: We thank the referee for their positive evaluation of our manuscript, insightful comments, and their support in making this manuscript more accessible. Herein, we provide a point-by-point response to the referee's comments. The changes in the manuscript have been tracked in blue. Following the referee's suggestions, we have also added an appendix section in the revised manuscript where we have reported the forecasting performance of T850 and a comparison with two WeatherBench models.

Referee comments:
1 Major Issues
1.1 U-STN may not be equivariance-preserving
1.1.1 Inconsistencies
It is now clear that the manuscript does not accurately describe the code that was used by the authors. Below are some inconsistencies that I have noticed:
[L147] "The parameters are learned through back-propagation": This could mislead readers into thinking that the 6 parameters of are uniquely learned through back-propagation, resulting in a single transformation (scaling+rotation+translation) enforced at evaluation time (after the network is trained). If I am not mistaken, this would not result in an equivariance-preserving network.

Authors' response: We agree with referee. We have revised the manuscript accordingly in Line 150 where we explicitly mention that the transformation between the input and output through the matrix $T(\theta)$ is equivariant and **not** the entire network. We have also revised Line 148 to indicate that the parameters θ are predicted for each sample.

Additionally, when looking at the code, I noticed that instead, are not trainable parameters but rather outputs of the last dense layer before up-sampling. Therefore, there is one matrix per sample, which is why the "transformations" tensor has the shape (Number of samples; 6) in the code of the (bilinear interpolation+affine transformation) layer.

[L153-155] Without further clarification, I fail to see how the above framework preserves SO (3) equivariance, which if I am not mistaken would here be defined along the lines of:

$$\forall\, \theta, \forall Inputs, U - STN[T_\theta(inputs) = \tau_\theta(U - STN(inputs))$$

Using the authors' encoding framework, wouldn't this be closer to enforcing robustness of U-STN's outputs to a range of pre-determined parameters regardless of the inputs, instead of learning the (scaling+rotation+translation) that maximizes accuracy for each sample separately?

Authors' response: We have explicitly mentioned in the revised manuscript that the entire network is not equivariant by construction in Line 154. Only the transformation in the latent space performed through $T(\theta)$ is used to capture translation, rotation, and scaling. We have also mentioned in Line 148 that the parameters θ are predicted for each sample.

[Figure 1] When looking at this schematic, it looks like the localization network output is transformed by T, resulting in the circled cross output that is then fed to the 5x5 convolutional kernel. Looking at the current version of the code, it looks like instead, the localization network output is used to produce a set of transformations T (Sample;), which is then applied to the bilinearly interpolated version of the original input Z(t), and not the input to the latent space as written in [L152]. I recommend addressing each one of these 4 inconsistencies separately, by either clarifying the manuscript or correcting its code.

Authors' response: The referee is correct in pointing out the bilinear interpolation requires the original input $Z(t)$. However, that is used to calculate the indices of the interpolation kernel. It is however then used on the latent space connected to the dense layers, similar to the original implementation of the spatial transformer network in Jaderberg et al., [1].

Referee's comment:

1.1.2 Confirming the superiority of U-STN over U-Net

The above inconsistencies made me wonder what exactly explains the accuracy gains of U-STN compared to its corresponding baseline U-Net. More specifically, to confirm that this superiority is indeed linked to the presence of the spatial
transformer module, would it be possible to:

1. Transparently communicate the number of learnable weights/biases/parameters in U-Net and U-STN? From the code, it looks like U-STN has 2; 461; 965 learnable parameters but I could not find the equivalent number for U-Net.
2. Additionally disclose the bottleneck size for U-Net and U-STN?

This would help affirm that U-STN performs better than U-Net because of the spatial transformer module and not simply because it has more learnable parameters or a larger bottleneck.

Authors' response: This is an interesting point raised by the referee. The performance improvement of U-STN and U-NET does not come from a difference in the number of parameters. While one can easily check the number of parameters within both the architectures using the

`model.summary()` API provided inside the codes, we would like to point out that both these architectures have separately underwent hyperparameter optimization where a much larger network for both U-NET and U-STN has been tested with even more number of filters . The architectures chosen in the manuscript are the most optimal ones based on significant trial and error. Both have roughly the same number of training parameters and it is important to note that U-NET with any larger number of parameters would perform poorly as compared to the reported performance. This is also true for U-STN. In both the architectures, the bottleneck layer has the same size of 8×16. We have reported this in the caption of Figure 1.

Referee's comment:

1.2 As a result, U-STN may not be physically constrained

[L43-45] U-STN is specifically introduced as a method to physically constrain a deep learning framework. Even if the authors use nuanced language, I find this motivation misleading for the reader, especially as both examples mentioned by the authors enforce invariance (which is different from equivariance, and even more different from the setup introduced in this manuscript): [2] weakly enforces invariants associated to the Navier-Stokes equations via the loss function, while [1] enforces the PDE structure (associated to invariants) and initial conditions as soft constraints and the boundary conditions as hard constraints in the case of the shallow water equations on a sphere. This is fundamentally different from the more data-driven and flexible approach adopted by the authors in this manuscript: Arguably, no physical constraints are enforced on the outputs of U-STN. Would it be possible to revise the introduction to clarify that this framework is not directly analogous to standard physics-informed approaches that can be found in the literature? Note that these revisions would not necessarily decrease the manuscript's impact but simply clarify the exact motivation (e.g., interpretability) and benefits (e.g., improved accuracy) of this novel architecture.

Authors' response: We agree with the referee's point here. The current set-up in this manuscript is very different from physics-informed architectures that enforces the exact PDE and boundary and initial conditions governing the system's dynamics. We thank the referee for this excellent suggestion. We have clarified this in the introduction of the revised manuscript and marked it with blue in Lines 75-76.

Referee's comment:

1.3 However, the current version of U-STN can help gain physical insight

While U-STN does not seem to enforce physical constraints, its strength could rely on the low-dimensional, interpretable construction of T. Once the authors confirm the superiority of U-STN over U-Net (1.1.2), it would be interesting to better understand how the transformation T_θ associated to each sample improves the accuracy of the data-driven forecast. This could be done quickly and within the manuscript's scope by uniquely decomposing each transformation T into its corresponding scaling, rotation, and translation (e.g., using this decomposition https://stackoverflow.com/questions/45159314/decompose-2d-transformation-matrix, or any

well-defined decomposition that the authors find interpretable). Once the transformation is decomposed, the authors could answer questions such as:

- How does the transformation vary from sample to sample (e.g., is the scaling, the rotation, or both different)?
- How does the transformation vary from field to field (e.g., does the transformation significantly change when adding temperature as an output)?
- What does this transformation mean physically (e.g., can the rotation or scaling be traced back to atmospheric wave properties)?
- Why does this transformation improve the accuracy of the data-driven forecast (e.g., are these just tunable parameters or is this transformation preventing erroneous assumptions about the rotational symmetry of e.g. wave breaking events as mentioned by the authors)?

This could be a good way to motivate the introduction of the spatial transformer module if it cannot be clearly proven that it preserves equivariance.

Authors' response: We appreciate the referee's comments on the interpretation of the transformation matrix $T(\theta)$. As we had indicated in our previous responses, we had tried to perform the decomposition of $T(\theta)$ for different samples as well as different fields following the referee's suggestion. While the parameters θ are different for different samples and different fields (Z500 and T850), it is hard to interpret and map the nature of transformation with respect to the change in the instantaneous fields especially since the transformation is performed on the low-dimensional space. We have carefully revised the manuscript and ensured that we do not make any claims about $T(\theta)$ capturing the correct representation of wavebreaking events in Lines 258-261 or that it captures any meaningful or interpretable transformations in Lines 264-267. The only evidence that we see in using $T(\theta)$ is an overall improvement in prediction performance.

We would also like to point out that in some of our ongoing work we have developed a more robust framework to perform physically meaningful interpretations of deep networks by using their spectral representations in Fourier space and can attribute the performance improvement of one architecture over another through the Fourier spectrum of the latent space. We intend to extend our framework to interpret the U-STN architecture as well and leave it for future work. We emphasize that the current set-up in this manuscript is difficult to interpret physically as mentioned in Lines 264-267 in the revised manuscript.

Referee's comment:

2 Minor Comments

[Table 1] (Very minor) Would "change appropriately in response to a transformation" be clearer than "change appropriately to a transformation" here?

Authors' response: Thank you for the suggestion. The revised manuscript has been updated.

[Figure 1] Would it be possible to specify the bottleneck's size on this schematic, i.e. the shape of the localization network output?

Authors' response: Thank you for the suggestion. While adding the information about the bottleneck size in Figure 1 makes it too busy, we have added the size of the bottleneck layer in the caption of the figure. We have also added the sizes of the outputs after each transformation in the architecture in Table 2.

[L198] Consider replacing R (\cal R) with R (\mathbb{R}) to follow conventions.

Authors' response: Thank you for the suggestion. The revised manuscript has been updated with this change.

[Equation 6] This looks like an auto-correlation matrix rather than a covariance matrix: Is this equation correct?

Authors' response: This is the equation that is used to calculate the background covariance matrix of the state $Z(t + 24\Delta t)$. A lagged autocorrelation matrix would have the two entries of the inner product differ in the "*lag"* value.

[L272] "not shown for brevity": As the manuscript is already rich in ideas and technical details, I understand the authors' motivation to only focus on one field in the main text. However, since the manuscript significantly gains in generality (from a meteorological perspective) by showing the improved prediction

Authors' response: We thank the referee for this suggestion. As we have shown in the previous responses (and in Figure 1 in this response), we have good prediction skill on T850 as well. We have added the figure comparing forecasting performance of T850 between U-STN12 and U-NET12 in Figure 10 in the revised manuscript in the appendix. However, in this manuscript, in order to integrate DA with DDWP and to introduce the multi-step DA framework we have taken Z500 as an example. As we have mentioned in the manuscript as well, we can use any other meteorological variable for this exercise.

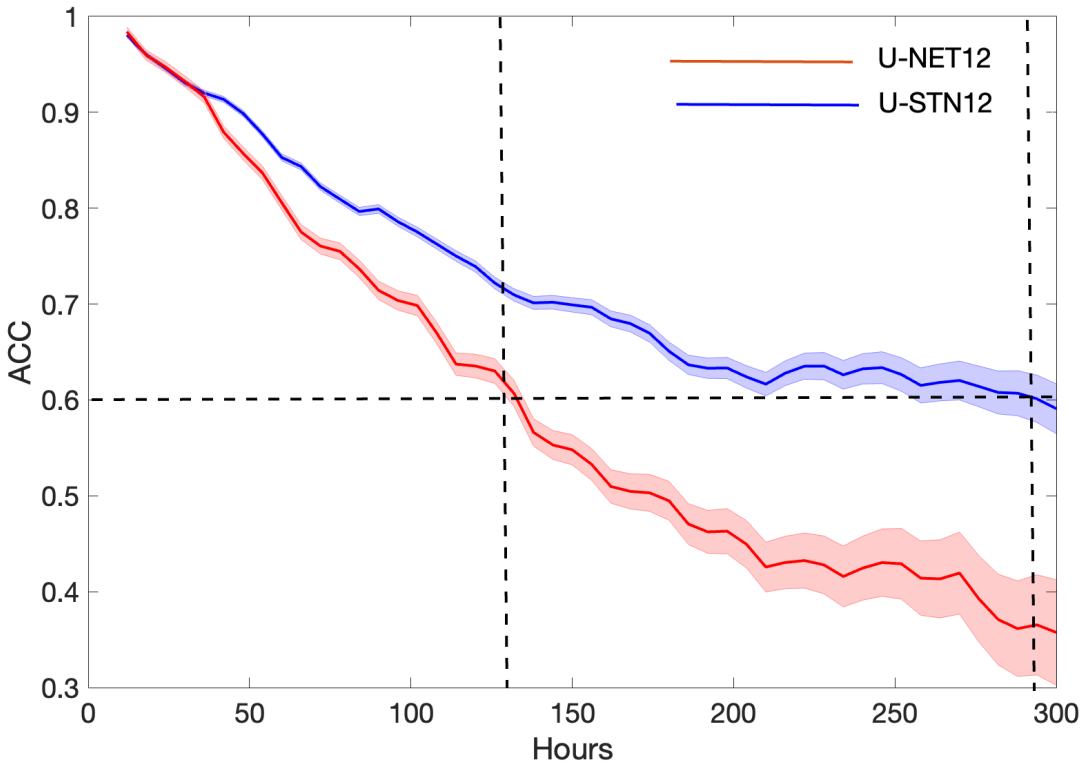

Figure 1. Anomaly correlation coefficient (ACC) calculated between T850 anomalies of ERA5 and T850 anomalies predicted using USTN12 and U-NET12 from 30 noise-free, random initial conditions. The solid lines and the shadings show the mean and the standard deviation over the 30 initial conditions.

[Figures 5,6,8,9] Missing units for RMSE (it should be meters if I am not mistaken).

Authors' response: We have mentioned the units of RMSE in the caption of each of the figures. Thank you for pointing this out.

[Caption of Figure 5] The authors use parentheses for clarification, abbreviations, references, and I find that using it to express opposites is confusing in this caption (see [4] for a general discussion on the topic). More specifically, it looks like the coefficient of determination R is used to abbreviate RMSE. Would it be possible to rewrite the caption to avoid potentially confusing the reader?

Authors' response: Thank you. We have re-written the caption.

[Figure 6, L275-280] Since it is difficult for readers to visualize numbers that are written in text, I highly recommend adding the WeatherBench baselines (and maybe [5, 3]) as scattered crosses on

the left plot of Figure 6. This would facilitate the comparison between the performance of these baselines and U-STN, which would further underline the good performance of U-STN.

Authors' response: We thank the referee for this suggestion. We had provided the comparison on RMSE in our previous response as well and we further provide that table in this response (Table 1). We have now added an appendix in the revised manuscript between Lines 390-391, where we have provided the comparison of RMSE between U-STN12 and two WeatherBench models in Table 3.

However, we do not intend this paper to be a comparison between different DDWP models on WeatherBench and other papers. Currently, WeatherBench has a leaderboard where we intend to submit our model for evaluation and comparison. But in this paper, we want to introduce DA to be integrated with DDWP and the novel multi-step DA framework.

Table 1. A comparison between U-STN12 presented in this manuscript and the linear regression and CNN models from WeatherBench [1]. 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**

[L320] Typo: "At 24th hours".

Authors' response: Thank you for pointing this out. We have now fixed it in the revised manuscript.

[L382] "forecast/assimilation": Do the authors mean "forecast and assimilation"? The current phrasing may confuse readers.

Authors' response: Yes, that is what we meant. We have now revised this line in the revised manuscript.

[Code availability statement] To clarify, I was referring to the authors' GitHub repository (and not the WeatherBench repository). Would it be possible to share the GitHub of this manuscript specifically (corresponding to the Zenodo URL https://zenodo.org/record/5553570#.YX-QKp5KhPY) as part of the code availability statement?

Authors' response: We thank the referee for this excellent suggestion. However, it seems that GMD does not allow us to put a Github link for our own code in the code availability statement and as soon as we submit, we receive our manuscript back for a revision that indicates this issue

with Github links which are not persistent archives. However, we intend to put the Github link of our codes on the arxiv pre-print. For the referee's ease of accessibility, we provide the Github link for our codes herein: https://github.com/ashesh6810/DDWP-DA.

References

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