Towards physically consistent data-driven weather forecasting: Integrating data assimilation with geometric deep learning in a case study with ERA5

First Round of Revisions

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.

Contents

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 θ are learned through back-propagation": This could mislead readers into thinking that the 6 parameters of $\boldsymbol{\theta}$ 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.

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

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

  $$\forall\boldsymbol{\theta}, \forall\mathbf{Inputs},\ \mathrm{U-STN}\left[\mathcal{T}_{\boldsymbol{\theta}}\left(\mathbf{Inputs}\right)\right] = \mathcal{T}_{\boldsymbol{\theta}}\left(\mathrm{U-STN}\left[\mathbf{Inputs}\right]\right).$$

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

- [Figure 1] When looking at this schematic, it looks like the localization network output is transformed by $\mathcal{T}_{\boldsymbol{\theta}}$, resulting in the circled cross output that is then fed to the 5×5 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 $\mathcal{T}_{(\mathrm{Sample},\boldsymbol{\theta})}$, 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.

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.

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.

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 $\mathcal{T}_{\boldsymbol{\theta}}$. 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 $\mathcal{T}_{\boldsymbol{\theta}}$ 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 $\mathcal{T}_{\boldsymbol{\theta}}$ 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.

2 Minor Comments

[Table 1] (Very minor) Would "change appropriately in response to a transformation" be clearer than "change appropriately to a transformation" here?
[Figure 1] Would it be possible to specify the bottleneck's size on this schematic, i.e. the shape of the localization network output?
[L198] Consider replacing $\mathcal{R}$ (\cal R) with $\mathbb{R}$ (\mathbb{R}) to follow conventions.
[Equation 6] This looks like an auto-correlation matrix rather than a covariance matrix: Is this equation correct?
[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
[Figures 5,6,8,9] Missing units for RMSE (it should be meters if I am not mistaken).
[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?
[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.
[L320] Typo: "At 24th hours".
[L382] "forecast/assimilation": Do the authors mean "forecast and assimilation"? The current phrasing may confuse readers.
[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?

References

[1] A. Bihlo and R. O. Popovych. Physics-informed neural networks for the shallow-water equations on the sphere. apr 2021.

[2] M. Raissi, A. Yazdani, and G. E. Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, feb 2020.

[3] S. Rasp and N. Thuerey. Data-Driven Medium-Range Weather Prediction With a Resnet Pretrained on Climate Simulations: A New Model for WeatherBench. *Journal of Advances in Modeling Earth Systems*, 13(2):e2020MS002405, feb 2021.

[4] A. Robock. Parentheses are (are not) for references and clarification (saving space), 2010.

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