Physics-Informed Neural Networks for Tropical Cyclone 2D Flow Reconstruction

Ryan Eusebi

23 December 2021

1 Introduction

1.1 Data Assimilation for Tropical Cyclones

In practice, the task of reconstructing the flow field of tropical cyclones (TCs) amounts to a data assimilation problem. Data assimilation involves taking sparse measurements recorded within a domain, and trying to reconstruct the flow of the fluid throughout the domain (both where you have recorded data, and where you don't). In doing so, one must rely on physical intuition of the system (such as the governing equations of motion or symmetry properties). Reconstructing full TC flows is important for a variety of reasons. The most important application is for initialization schemes for dynamical forecast models - in order to accurately forecast the evolution of the TC using numerical modeling, we need an accurate picture of what the full TC wind and pressure fields look like at our initial time steps in the forecast model. Cavallo et al. (2013) discuss how inadequate initialization schemes are largely responsible to forecast errors for TCs [2]. This is especially true of intensity forecasts (predicting the maximum winds and minimum central pressure of storms at various times in the future), which especially suffer from inadequate initialization [2, 9]. In addition to improving forecast skill, reconstructing TC flows would better facilitate research on impacts of winds from historical hurricanes, and would allow for better insurance risk assessment and augment our understanding of climatological TC impacts.

Data assimilation for TCs is especially difficult because the wind and pressure fields are so large in both the horizontal (hundreds of kilometers) and vertical scales (kilometers), and assimilation methods are heavily constrained by the small amount of real-time recorded measurements from within storms. One common source of data available for real-time forecasting is hurricane hunter data, which come from planes that fly horizontal transects through the eye of the storm at a flight level of anywhere between the 850hPA and 700hPa pressure levels. These planes can measure the 3d winds and pressure (among other variables) along the paths they trace out [3]. These planes also drop small instruments called dropsondes throughout their flights through the storm. These tend to be dropped in the eye of the storm where the strongest winds and most significant variability are, and they provide real-time vertical wind and pressure profiles as the dropsonde quickly drops from the plane to the ocean surface [3, 5]. Additional forms of available data might include buoy measurements recorded at the ocean surface underneath the TC.

In more recent years, enhanced satellite technology (such as the GOES-16 satellite) has offered newer high-resolution data which can be incorporated into assimilation schemes. For instance, satellite estimates of atmospheric motion vectors (AMVs) are commonly being utilized in the newest methods [4]. AMVs are estimates of wind vectors generated by comparing how cloud and water vapor structures change from one satellite image to the next, and the increased spatial and temporal resolutions of new satellites have enabled very accurate readings [10]. However, these wind vectors can only be calculated at certain finite pressure levels (historically this used to be 850hPa and 200hPa, because that is where cumulous and cirrus cloud tops were respectively, but newer satellites allow for measurements at more heights), and they aren't totally accurate [10].

1.2 Previous Approaches

There have been various approaches to the data assimilation problem. Early attempts at reconstructing hurricane 2D and 3D profiles relied on the assumption of a symmetric vortex in gradient wind balance (the pressure gradient force of the cyclone balanced by the coriolis and centrifugal forces) [1, 6]. However, these methods are known to be inaccurate [1], and many new methodologies have been developed since those early attempts. Most newer methodologies involve starting with an initial guess of the vortex, but then iteratively adjusting the field based on the equations of motion used by the dynamical model until your initialization somewhat matches the observations you have [2]. This is a very complicated and inefficient process, and prior studies have highlighted its weaknesses [2]. The newest models also incorporate numerous sources of data, including satellite measurements of AMVs, but they struggle to efficiently incorporate vast amounts of data into the assimilation system [4].

1.3 Physics-Informed Neural Networks

Physics-informed neural networks (PINNs) are a new machine learning methodology which aims to combine physical knowledge of a system with the power and robustness of the popular machine learning algorithm, neural networks (NNs). They operate very similar to a normal NN, except their loss function includes the partial differential equations (PDEs) that govern the system which you are trying to model (this is possible since NNs are differentiable functions, and their partial derivatives with respect to the outputs can be readily calculated and used in the loss function to motivate the output to satisfy the relevant PDEs). Section 3.3 will provide more details about how PINNs work.

PINNs have found much success in a variety of applications in recent years. They have proved themselves to be especially useful in ”inverse problems,” when observations of some component of a flow field are sparsely recorded, and the PINN is tasked with reconstructing the full flow field. Inverse problems are essentially data assimilation problems like the task of data assimilation for TCs. Raissi et al. (2018) built a PINN which was able to accurately recover wind and pressure fields using sparse observations and the Navier-Stokes (conservation of momentem) and continuity (conservation of mass) equations for different 2D and 3D flows [8]. Zhang and Zhao (2021) built a PINN for small-scale low-magnitude turbulent winds on a solar farm [11]. The PINN showed promising results, as it was able to recover the 2D spatio-temporal variability of the flow of the system using very spatially sparse measurements. There are many other applications of PINNs, but very few if any are applied to atmospheric and/or large-scale flows (where the coriolis force becomes significant), and there are especially no examples of PINNs being applied to TC data assimilation.

2 Problem Statement

The aim of this project is to construct a PINN capable of reproducing 2D hurricane wind flows, including the wind (both the zonal and meridional components) and pressure fields. The PINN should be able to recover the spatial and temporal variability of the flow. Given the 2D constraint of our objective, we will focus on a single pressure level around the 800 hPa pressure level, which is high enough so that the effects of friction can be ignored and hurricane hunters would be able to fly at that altitude and collect reliable data. The PINN will be trained for its output to match observations we have collected within the storm and to also satisfy the relevant PDEs throughout the output domain which govern fluid flow for TCs.

The main goal of the project is to determine how effective the PINN methodology is at reconstructing hurricane wind flows. Another key aim is to determine what combinations and patterns of observational data are necessary and bettter for optimal PINN results. Determining which equations of motions to use (and in what form) is also important, as well as figuring out the hyperparameters for the PINN (e.g. how should the equation and data losses be weighted). These goals will be assessed using both quantitative and qualitative metrics.

3 Methodology

3.1 Data

Since accurate and complete real-world data of TC spatial and temporal wind fields generally doesn't exist, numerical modeling data is used for the purposes of this study. Namely, the AM2.5C360 25-km resolution atmospheric climate model from the Geophysical Fluid Dynamics Laboratory is utilized. This model has been shown to accurately reproduce climatological TC features and frequencies, including hurricane intensities (it is able to capture the dynamics and frequencies of intense storms, unlike lower resolution models) [7]. For this study, we reran one year of the model forced with observed 2020 sea surface temperatures, and saved instantaneous (as opposed to time-averaged) data at a 3-hour frequency for numerous pressure levels between 1000 and 5 hPa. We assume that the physics in these models accurately represents the physics in our real-world system.

Of the available pressure levels from the model, we choose the 811 hPa pressure level for our 2D analysis. This level is high enough such that the effects of friction can largely be ignored in the momentum equations (see section 3.2), and is also at a height at which hurricane hunter planes can fly and record data. For the purposes of this project, we largely focus on a case study of a single category 4 storm observed within the model. The storm was located in the Philippine Sea in the western North Pacific with maximum surface-level winds of 62 m/s. We refer to this storm as H4-PHI throughout the paper. We also provide a brief quantitative overview on the results of the PINN on various other storms, too, which are arbitrarily chosen among the available storms from the year of modeling. For modeling storms, we make use of 3 time steps of data of the storm available from the climate model, each 3 hours apart. The times of these 3 steps for a given storm are labeled as $t = \{-3, 0, 3\}$ where the units are hours. The wind fields for H4-PHI at those 3 time steps are shown in Figure 1. Using those 3 time steps of data, we attempt to reconstruct the flow for the time domain $t \in [-4.5, 4.5]$. So the output of our PINN will not only reconstruct the flow at the times $t = \{-3, 0, 3\}$, but will also both extrapolate and interpolate beyond and in between those times. Our output domain is an 800km x 800km (at 20km resolution) grid centered around the storm center at time $t = 0$.

The PINN needs data points, which are measurements recorded within the storm you are trying to model, as input for training. We simulate measuring data points by sampling certain datapoints from the grid at the times for which we have data (Figure 1). We experiment with three different patterns of data points for this study, inspired by existing hurricane hunter flight plans for TCs. We refer to the first as the "All" data pattern, which is unrealistic but serves as a sanity check for our model - we simply sample all datapoints in the 800x800km grid around the storm for all 3 time stamps. The next data pattern is the

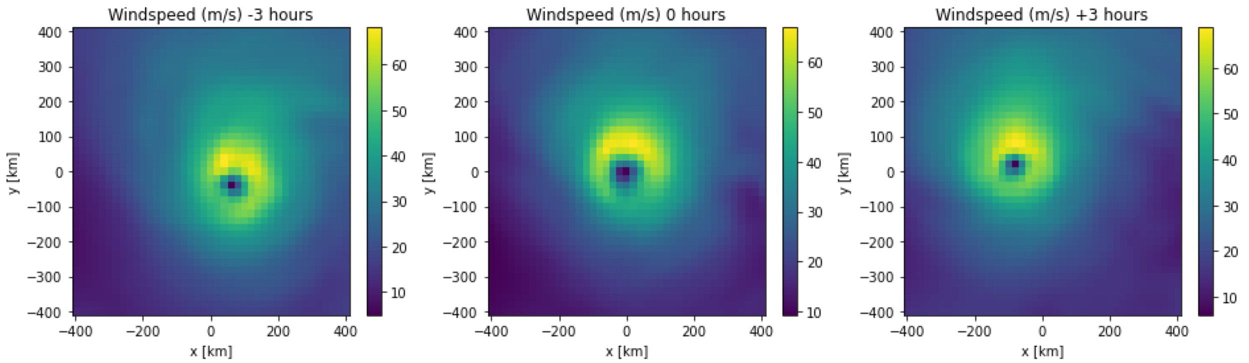

Figure 1: Category 4 storm in the Philippine Sea at times $t = -3$ (left), $t = 0$ (center), and $t = 3$ (right) where the units for time are hours. We see the storm is moving west-northwest.

”Double Cross” pattern, which consists of 4 passes through the eye of the storm. The final data pattern is the ”Cross” pattern which consists of 2 passes through the eye of the storm. For the ”Double Cross” and ”Cross” patterns, we also use the boundary gridpoints as data points for the model, since we assume these will be low in magnitude and can easily be guessed or determined by prior numerical modeling. The three patterns are shown in Figure 2 as they would be applied to a single time point of a storm.

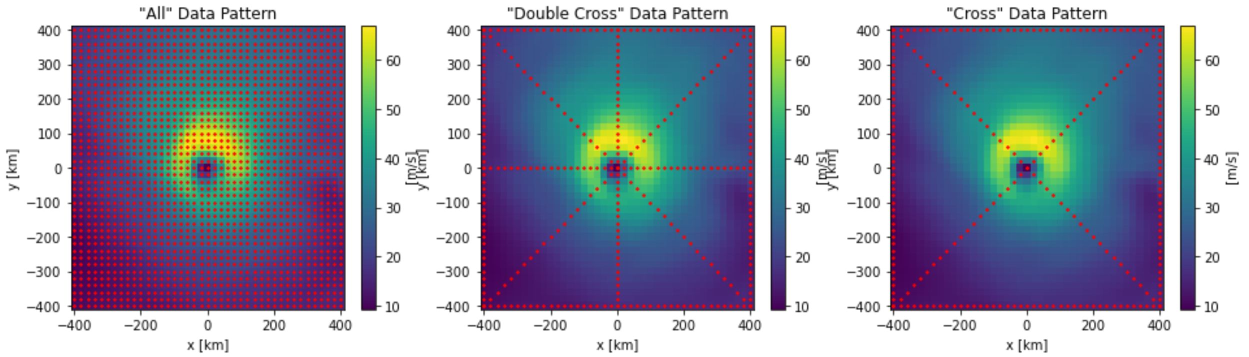

Figure 2: Three different data point patterns used in this paper: ”All” (left), ”Double Cross” (center), and ”Cross” (right). The red points represent the points which were used as measured data points for the respective patterns. This storm is referred to extensively throughout this study.

3.2 Equations of Motion

For this study, we rely on the horizontal Navier-Stokes equations for large-scale flows on a rotating planet (presented in Equations 1a and 1b) and the continuity equation (presented in Equation 2). In constructing these equations, we assume both hydrostatic balance (shown in equation 3) and non-divergent flow (which results in our continuity equation in equation 2). We also assume friction is negligible at the 811hPa level, thus resulting in no friction term in our momentum equation. These approximations are not exactly true for our system, which has very high windspeeds and stronger vertical winds than are seen in other systems, but the approximations are necessary to simplify our set of equations. Note that since the data available to us is along surfaces of constant pressure, not height in meters, our equations are also for motion along a constant pressure surface. The vertical coordinate becomes p for pressure, instead of the usual z for height. The vertical velocity also becomes $\omega = \mathrm{d}p/\,\mathrm{d}z$ instead of the usual $w = \mathrm{d}z/\,\mathrm{d}t$. Using pressure as the vertical coordinate and the the hydrostatic approximation allows us to write the pressure term as seen on the right hand side of equations 1a and 1b, with $\Phi = \rho g$ being the geopotential height with units $\mathrm{m}^2/\mathrm{s}^2$, representing the height of the pressure surface in meters multiplied by the gravitational constant. Pressure coordinates with the hydrostatic approximation causes the density term to drop out from the momentum equations (the pressure term when using vertical coordinates would be $-\frac{1}{\rho}\frac{\partial p}{\partial z}$), which means there is one less parameter we need to worry about calculating/measuring. Note that we also use the beta-plane approximation for the coriolis term (the last term on the left hand side of equations 1a and 1b) where f_o is the coriolis parameter at the latitude of the storm center at time $t = 0$ and β is the constant $\beta = \partial f/\partial y$.

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \omega\frac{\partial u}{\partial p} - (f_o + \beta y)v = -\frac{\partial \Phi}{\partial x} \tag{1a}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \omega\frac{\partial v}{\partial p} + (f_o + \beta y)u = -\frac{\partial \Phi}{\partial y} \tag{1b}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial \omega}{\partial p} = 0 \tag{2}$$

$$\frac{\partial p}{\partial z} = -\rho g \tag{3}$$

Another important point regarding these equations is that despite us only modeling the 2D flow at one pressure level, the equations we present in this section are for 3D flow. We explain in section 3.3 how despite only modeling the 2D flow, it was necessary to include the vertical coordinate terms in the equation, too.

Note that in these equations, and for the rest of the paper, (in addition to the symbols already described) u (v) is the zonal (meridional) wind component along a constant pressure surface, and x (y) is the spatial coordinate in the zonal (meridional) direction. So $u = \mathrm{d}x/\,\mathrm{d}t$ and $v = \mathrm{d}y/\,\mathrm{d}t$.

3.3 PINN Structure

Figure 3 illustrates the final structure of the PINN used in this study. We see that the 4 inputs include the 3 spatial coordinates (x, y, and p) and the time coordinate (t). The outputs are the 3 components of the

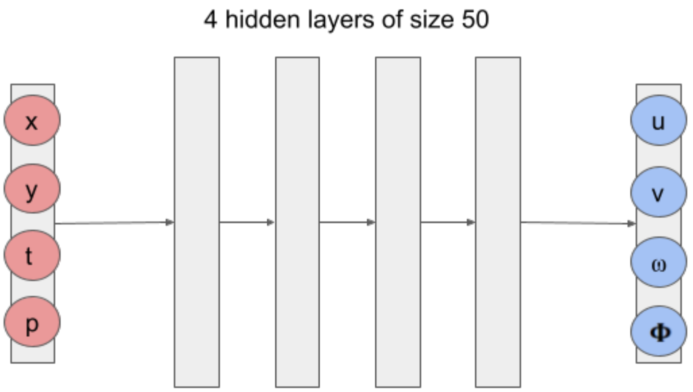

Figure 3: Structure of PINN used in this study, indicating the inputs and outputs and the 4 hidden layers, each with 50 nodes.

wind vector (u, v, and ω) and the geopotential height (Φ) at the input coordinates. So the PINN essentially approximates a solution to the Navier-Stokes and continuity equations in the domain we are concerned with, and allows us to calculate the wind and pressure fields at any point that is inputted into the model. The four fully connected hidden layers and the output layer each have a bias layer - the biases combined with the matrices used to map each layer to the next result in a total of 8,104 learnable parameters for the entire PINN. Even though we are only trying to reconstruct the 2D flow of the TC, we still include the vertical coordinate, pressure, as an input and the vertical winds, ω, as an output. This is because while training, we found that the vertical terms in the momentum balance were important for the equations to balance, so we created an "artificial" pressure coordinate - we motivated the model to learn what the vertical winds and vertical gradients should be despite not giving any vertical information to the model. In our experiments, this enhanced the quality of the model.

In order to train the model, we need a loss function. But before we can define the loss function, we need to define what the inputs to our model for training will be. A PINN requires data points and collocation points for training. Data points are measurements recorded within the storm (for our purposes, data points sampled from the climate model output), while collocation points are points within the domain of the flow at which we evaluate our governing equations of motions (equations 1a, 1b, and 2) to ensure the model's output is reproducing a flow that is physically realizable. So there will be two components to our loss function: the loss from the data points and the loss from the collocation points. At the data points, we have measured values of u, v, and Φ (note that we don't have ω since vertical winds are not included in the model output), so the contribution from the data points to the loss function (referred to as MSE_{Data}) is defined by equation 4. The subscript "pred" indicates the values predicted by the PINN, and the subscript "actual" indicates the actual measured values sampled from the climate model data. Since we don't have measurements at the collocation points, their contribution to the loss function comes strictly from how closely the equations of motions are satisfied. This loss component (referred to as $\text{MSE}_{\text{Equation}}$) is defined by equation 5. The summation in equations 4 and 5 indicates summation over all n data points and m collocation points respectively. Note that in the code, the equations are non-dimensionalized so that each variable ranges between -1 and 1 - this removes effects from different units and allows each term to be equally important in the loss function.

$$\text{MSE}_{\text{Data}} = \frac{1}{n}\sum\left[(u_{\text{pred}} - u_{\text{actual}})^2 + (v_{\text{pred}} - v_{\text{actual}})^2 + (\Phi_{\text{pred}} - \Phi_{\text{actual}})^2\right] \tag{4}$$

$$\begin{aligned}\text{MSE}_{\text{Equation}} = \frac{1}{m}\sum\Bigg[&\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \omega\frac{\partial u}{\partial p} - (f_o + \beta y)v + \frac{\partial \Phi}{\partial x}\right)^2 \\ &+ \left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \omega\frac{\partial v}{\partial p} + (f_o + \beta y)u + \frac{\partial \Phi}{\partial y}\right)^2 \\ &+ \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial \omega}{\partial p}\right)^2\Bigg]\end{aligned} \tag{5}$$

Recall that since a PINN is a differentiable function, the partial derivatives in equation 5 can readily be calculated from the outputs. The total loss function used by the PINN is defined in equation 6 where γ is a hyperparameter which controls how the equation loss is weighted relative to the data loss.

$$\mathcal{L} = (1-\gamma)\text{MSE}_{\text{Data}} + \gamma\text{MSE}_{\text{Equation}} \tag{6}$$

4 Results

4.1 PINN Training

The training loss curves when training on H4-PHI using our three different data point patterns is shown in Figure 4. In all cases, the equation loss drops lower than the data loss, and the equation loss usually falls very fast early on in the training process. When the "All" pattern is used, the total loss is the lowest of the three patterns and the gap between equation and data loss is the biggest, too, which suggests that our system of equations doesn't perfectly capture the dynamics of the system. However, the loss curves get low enough that we assume the equations to be good enough to get useful results.

Models were trained for 20,000 iterations with 20,000 collocation points - training time took around 40 minutes using 40 CPU cores. The 20,000 collocation points were randomly generated within the domain of our system, with the pressure bounds being +/- 25 hPa of 811 hPa (the pressure level we want predictions at). While they were randomly generated, we designed it so that half of the points would be in the 400km x

400km center of the grid (where the eye of the storm and the strongest winds and variability are), and the other half of the points were in the rest of the 800km x 800km grid. We found it was best to weight the data loss more than the equation loss, otherwise the equation loss would decrease very sharply in the beginning and not allow the data loss to come down too much, so $\gamma = 0.1$ was used. It was found that using more collocation points, training for more iterations, and using larger or more hidden layers did not improve the results and only increased the training time.

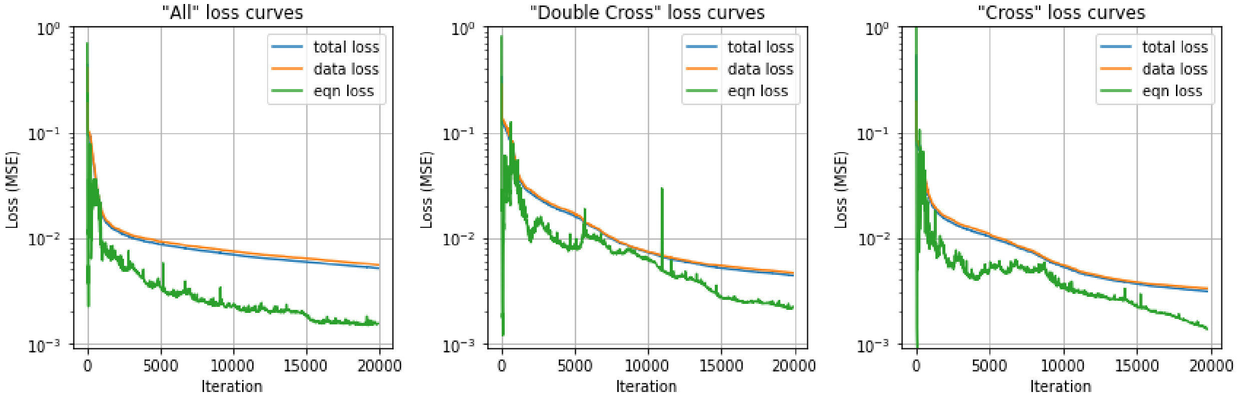

Figure 4: Loss curves for training on H4-PHI using the ”All” (left), ”Double Cross” (center), and ”Cross” (right) data patterns.

4.2 PINN Outputs

Figure 5 shows output wind speed fields at time $t = 0$ for the H4-PHI storm. The top row in the figure is the output, and the bottom row is the error (difference between PINN wind speeds and actual wind speeds). Each column uses a different data pattern in training. We see that the ”All” data pattern has very small errors, with just a spike in error on the southern side of the eyewall where it overestimated the winds. For the ”Double Cross” pattern, the results are a little worse, as expected, but overall still very good as errors around the eyewall generally top out around 5-6 m/s. The ”Cross” pattern has a few extreme errors around the eyewall (likely in the regions it doesn't have data), but overall still does a good job at capturing the large-scale features of the field. The PINN works similarly well for the wind field at times $t = -3$ and $t = 3$. The PINN also offers very reasonable interpolations and extrapolations for the storm, too. We have no ground truth data to compare against other than the data at times $t = -3, 0, 3$, but the interpolations and extrapolations continue moving the center of the storm in the direction it is supposed to be moving, and they hold together the structure and illustrate how the flow evolves from one time step to the next. The extrapolations start to fall apart at the endpoints $t = -6$ and $t = 6$.

We also applied the model to various storms of different intensities, and the quantitative results are

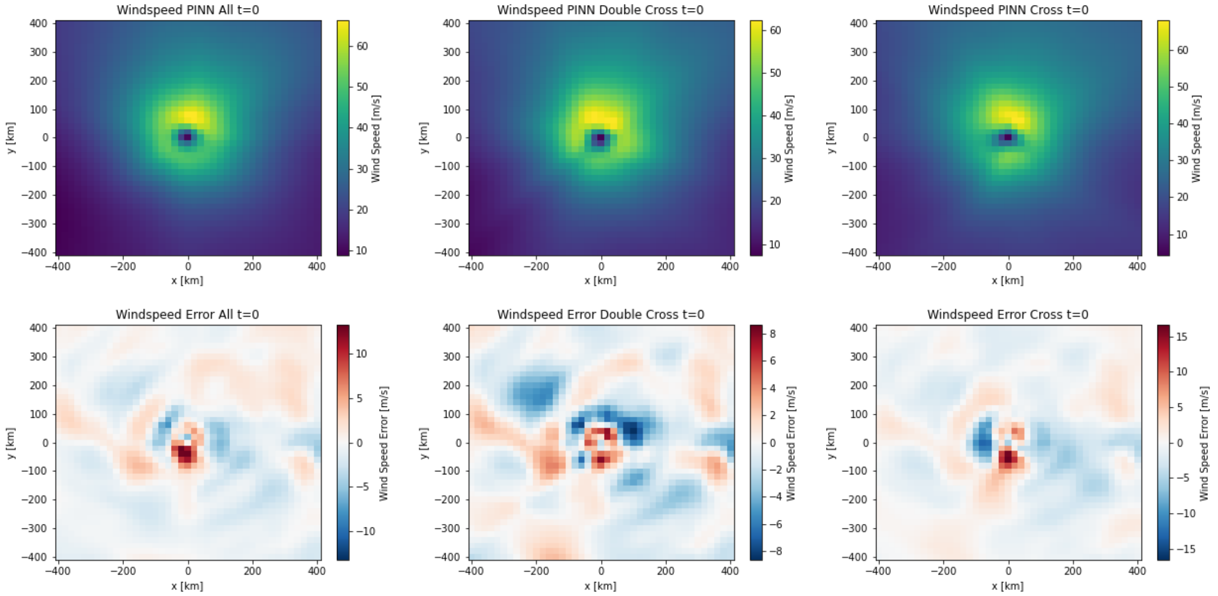

Figure 5: Outputs of the PINN for H4-PHI. From left to right, we have outputs for the "All," "Double Cross," and "Cross" data patterns. The top row is the wind speed from the output of the PINN for $t = 0$ (formed by combining the u- and v-component winds). These outputs can be compared against the ground truth $t = 0$ plot in Figure 1. The bottom row is the error (PINN output minus the ground truth wind field).

summarized in Tables 1 and 2. Each table provides the absolute value error (ABS) in m/s and mean square error (MSE) for the PINN at time $t = 0$ for each storm for the three different data patterns the PINNs were trained with. Table 1 averages the error over the entire 800x800km grid, while Table 2 averages just over the 400x400km grid centered around the center of the storm. Since the strongest winds and highest variability occurs near the storm's center, Table 2 gives an overview of how well the model is doing at reconstructing those strong, turbulent winds. Overall, the PINN seems to be doing a very good job with the whole wind field and the winds near the eye when it uses either the "All" or "Double Cross" data pattern, but losses are very high when it uses the "Cross" pattern. The PINN's effectiveness varies from storm to storm, but is generally consistent. As is expected, the loss tends to be lower for weaker storms.

		All		Double Cross		Cross	
Storm	Max Wind (m/s)	ABS	MSE	ABS	MSE	ABS	MSE
1 (H4-PHI)	62	1.28	3.29	1.35	3.56	1.59	5.45
2	27	0.57	0.55	1.07	1.92	1.49	7.26
3	32	0.52	0.50	0.89	1.65	1.66	7.67
4	45	1.27	4.63	1.50	4.46	2.40	12.9
5	25	0.81	1.26	1.33	3.12	1.38	5.90
6	40	0.79	1.45	1.32	3.49	1.73	6.30

Table 1: Results for 6 different storms with the three different data patterns being used on each. Absolute value loss (ABS) in m/s and mean square error loss (MSE) averaged across the entire 800x800km grid for $t = 0$. The max wind column indicates the maximum surface-level winds.

		All		Double Cross		Cross	
Storm	Max Wind (m/s)	ABS	MSE	ABS	MSE	ABS	MSE
1 (H4-PHI)	62	2.06	8.27	2.23	8.35	2.79	14.4
2	27	0.63	0.67	1.34	2.99	2.61	19.2
3	32	0.74	0.95	1.33	3.19	3.25	20.1
4	45	2.30	13.7	2.56	10.7	4.56	33.0
5	25	0.91	1.50	1.68	4.86	2.78	22.6
6	40	1.20	3.50	1.78	5.73	2.99	15.1

Table 2: Results for 6 different storms with the three different data patterns being used on each. Absolute value loss (ABS) in m/s and mean square error loss (MSE) averaged across the 400x400km grid centered around the storm center for $t = 0$. The max wind column indicates the maximum surface-level winds.

5 Discussion and Directions for Further Research

We see that the PINN is effectively able to reconstruct the flow field of the TC, but that its success was very contingent on the amount of data / the pattern used for the observed data points. Under the "All" and "Double Cross" data patterns, it proved very effective, and allowed for high spatial and temporal resolution interpolations and reasonable extrapolations, too. The "Double Cross" pattern results, however, did have some artifacts in their outputs, and the "Cross" pattern results were very inadequate. The results demonstrate that a PINN is a promising alternative to existing data assimilation techniques, and it certainly warrants further research.

Building on the 2D case explored in this study, there are many avenues for further research. One next step would be to explore how different data point patterns affect the results (the data patterns used were modeled after existing hurricane hunter flight paths, but maybe different paths would be more advantageous for the PINN's training). Additionally, methods could be explored to enhance the effectiveness of the model for the existing data patterns with which we experimented. For instance, techniques which could promote radial symmetry of the wind speed about the center of the storm could be explored to motivate the model to find the solution to the equations which most resembles a real-life TC (we saw that when using the "Cross" pattern, the PINN outputted some extreme values in between gaps in data points where we know the values should be similar in magnitude to the data points at a similar radius). Finally, more testing is needed to determine how well the model is doing at both interpolating and extrapolating.

Beyond the 2D case, the PINN should definitely be applied to the full 3D case to attempt to reconstruct the full 3D flow throughout the entire vertical domain. This would utilize the same equations we presented in section 3.2, but we would include data points from all pressure levels in a manner that is realistic to what would be measured by dropsondes and satellites. For the 3D case, a vertical momentum equation might need to be added to the equation loss - this might necessitate the PINN having to calculate the density field of the flow, too. Since density is determined by the ideal gas law for moist air, density can be calculated from specific humidity and temperature, so those two variables might need to be added to the output of the PINN as well (and these are two variables which are typically recorded with real-time observations along with wind and pressure).

Eventually, experiments can be done with the PINN to assess how well it does on reconstructing the flows of actual hurricanes, using actual observations measured from aircraft, dropsondes, and satellites. The best way to assess the PINN in this context would be to generate the 3D vortex based on the recorded measurements, and stitch that vortex into a dynamical forecast model. The accuracy of the forecast can then be compared against the accuracy when other initalization schemes are used to determine how the PINN performs relative to the other schemes.

References

[1] R. A. Anthes, "Data assimilation and initialization of hurricane prediction models," *Journal of Atmospheric Sciences*, vol. 31, no. 3, pp. 702 – 719, 1974. [Online]. Available: https://journals.ametsoc.org/view/journals/atsc/31/3/1520-0469_1974_031_0702_daaioh_2_0_co_2.xml

[2] S. M. Cavallo, R. D. Torn, C. Snyder, C. Davis, W. Wang, and J. Done, "Evaluation of the advanced hurricane wrf data assimilation system for the 2009 atlantic hurricane season," *Monthly Weather Review*, vol. 141, no. 2, pp. 523 – 541, 2013. [Online]. Available: https://journals.ametsoc.org/view/journals/mwre/141/2/mwr-d-12-00139.1.xml

[3] J. Collins and P. Flaherty, "The noaa hurricane hunters: A historical and mission perspective," *Florida Geographer*, vol. 45, pp. 14–27, 01 2014.

[4] R. L. Elsberry, J. W. Feldmeier, H.-J. Chen, M. Peng, C. S. Velden, and Q. Wang, "Challenges and opportunities with new generation geostationary meteorological satellite datasets for analyses and initial conditions for forecasting hurricane irma (2017) rapid intensification event," *Atmosphere*, vol. 11, no. 11, 2020. [Online]. Available: https://www.mdpi.com/2073-4433/11/11/1200

[5] J. L. Franklin, M. L. Black, and K. Valde, "Gps dropwindsonde wind profiles in hurricanes and their operational implications," *Weather and Forecasting*, vol. 18, no. 1, pp. 32 – 44, 2003. [Online]. Available: https://journals.ametsoc.org/view/journals/wefo/18/1/1520-0434_2003_018_0032_gdwpih_2_0_co_2.xml

[6] Y. Kurihara, M. A. Bender, and R. J. Ross, "An initialization scheme of hurricane models by vortex specification," *Monthly Weather Review*, vol. 121, no. 7, pp. 2030 – 2045, 1993. [Online]. Available: https://journals.ametsoc.org/view/journals/mwre/121/7/1520-0493_1993_121_2030_aisohm_2_0_co_2.xml

[7] H. Murakami, G. A. Vecchi, S. Underwood, T. L. Delworth, A. T. Wittenberg, W. G. Anderson, J.-H. Chen, R. G. Gudgel, L. M. Harris, S.-J. Lin, and F. Zeng, "Simulation and prediction of category 4 and 5 hurricanes in the high-resolution gfdl hiflor coupled climate model," *Journal of Climate*, vol. 28, no. 23, pp. 9058 – 9079, 01 Dec. 2015. [Online]. Available: https://journals.ametsoc.org/view/journals/clim/28/23/jcli-d-15-0216.1.xml

[8] M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden fluid mechanics: A navier-stokes informed deep learning framework for assimilating flow visualization data," 2018.

[9] E. N. Rappaport, J. L. Franklin, L. A. Avila, S. R. Baig, J. L. Beven, E. S. Blake, C. A. Burr, J.-G. Jiing, C. A. Juckins, R. D. Knabb, C. W. Landsea, M. Mainelli, M. Mayfield, C. J. McAdie, R. J. Pasch, C. Sisko, S. R. Stewart, and A. N. Tribble, "Advances and challenges at the national hurricane center," *Weather and Forecasting*, vol. 24, no. 2, pp. 395 – 419, 2009. [Online]. Available: https://journals.ametsoc.org/view/journals/wefo/24/2/2008waf2222128_1.xml

[10] C. S. Velden and J. Sears, "Computing deep-tropospheric vertical wind shear analyses for tropical cyclone applications: Does the methodology matter?" *Weather and Forecasting*, vol. 29, no. 5, pp. 1169 – 1180, 2014. [Online]. Available: https://journals.ametsoc.org/view/journals/wefo/29/5/waf-d-13-00147_1.xml

[11] J. Zhang and X. Zhao, "Spatiotemporal wind field prediction based on physics-informed deep learning and LIDAR measurements," *Applied Energy*, vol. 288, no. C, 2021. [Online]. Available: https://ideas.repec.org/a/eee/appene/v288y2021ics0306261921001732.html