

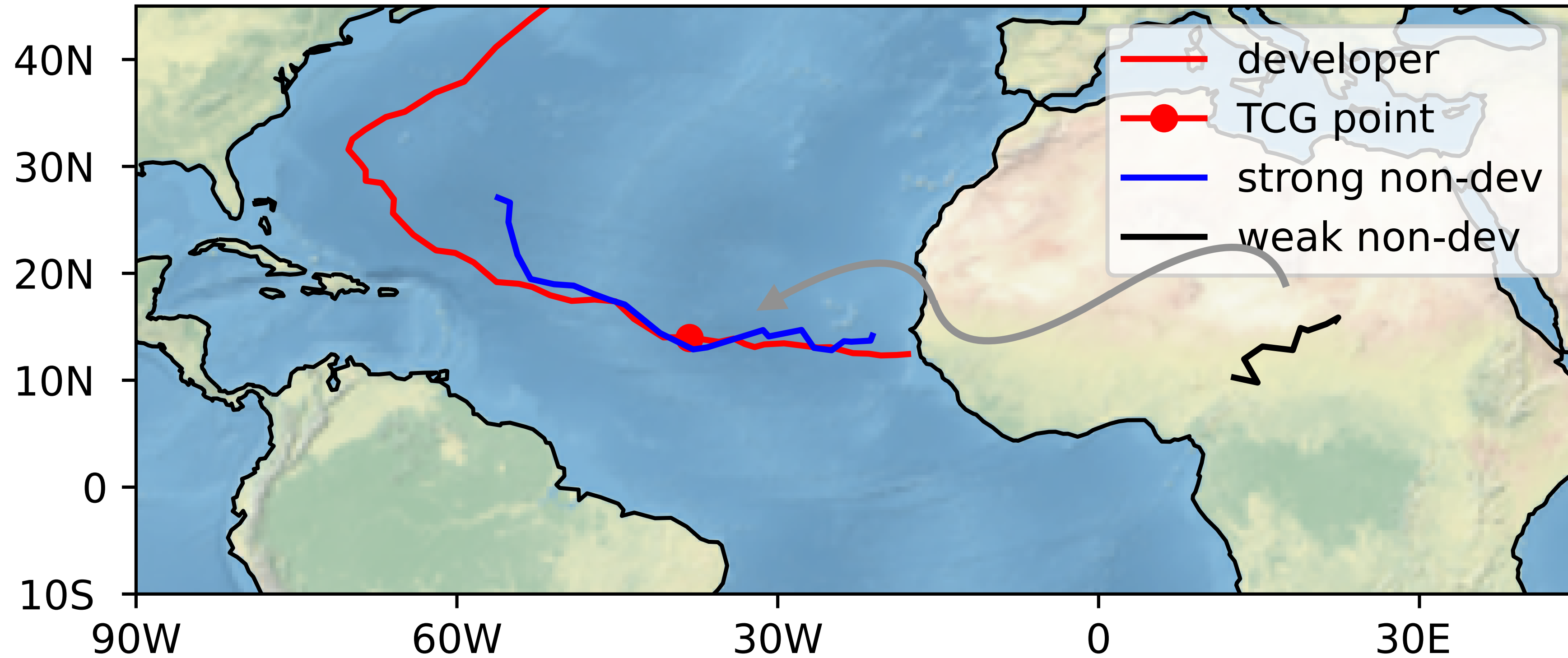
Machine Learning Approaches on Identifying Tropical Waves That Develop into Hurricanes

Haochang Luo¹, Spencer A. Hill^{1,2}

Mentors from CCAI mentorship program: Manmeet Singh³, Naveen Sudharsan³

*¹Department of Earth and Atmospheric Sciences, City College of New York, New York, NY ; ²Lamont-Doherty Earth Observatory, Columbia University, New York, NY; ³Department of Earth and Planetary Sciences, Jackson School of Geosciences, University of Texas, Austin, TX.
email: hluo@ccny.cuny.edu*

Background & theories



- African Easterly Waves (AEWs) are westward-propagating storms that grow from the instabilities in the African Easterly Jet (grey curve). They are weaker than hurricanes.
- ~85% of major hurricanes (Categories 3-5) start out as AEWs and then develop into hurricanes. But only 3% of AEWs develop into hurricanes.
- Whether a given AEW will develop into a hurricane or not remains hard to predict.

So can machine learning (ML) approaches be used to differentiate developing and non-developing AEWs?

Classification problem, computer vision task

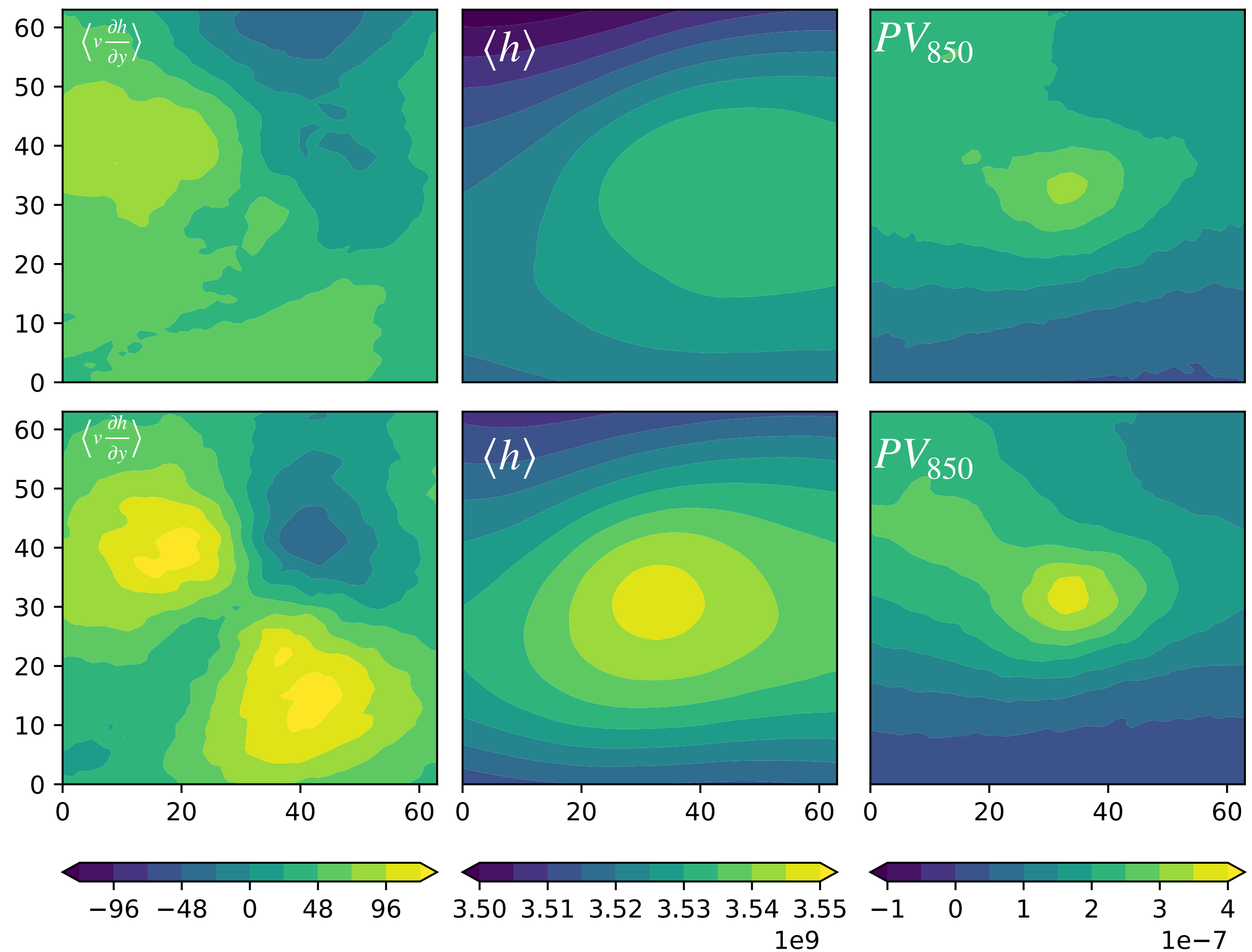
three key variables (RGB channels):

1. Column-integrated moist static energy (MSE): sum of sensible heat, geopotential and latent energy, integrated from top to bottom
2. Column-integrated meridional advection of MSE (Moisture-vortex instability theory): $\left\langle v \frac{\partial h}{\partial y} \right\rangle$
3. Potential vorticity at 850 hPa

Datasets

- The fifth reanalysis product from the European Centre for Medium-Range Weather Forecasts reanalysis data (ERA5). 1979-2018
0.25° × 0.25° horizontal resolution, 6-hour temporal resolution
- Storm tracking dataset: latitudes & longitudes of storm centers (developers and non-devs), storm types. (Cheng et al. 2019)

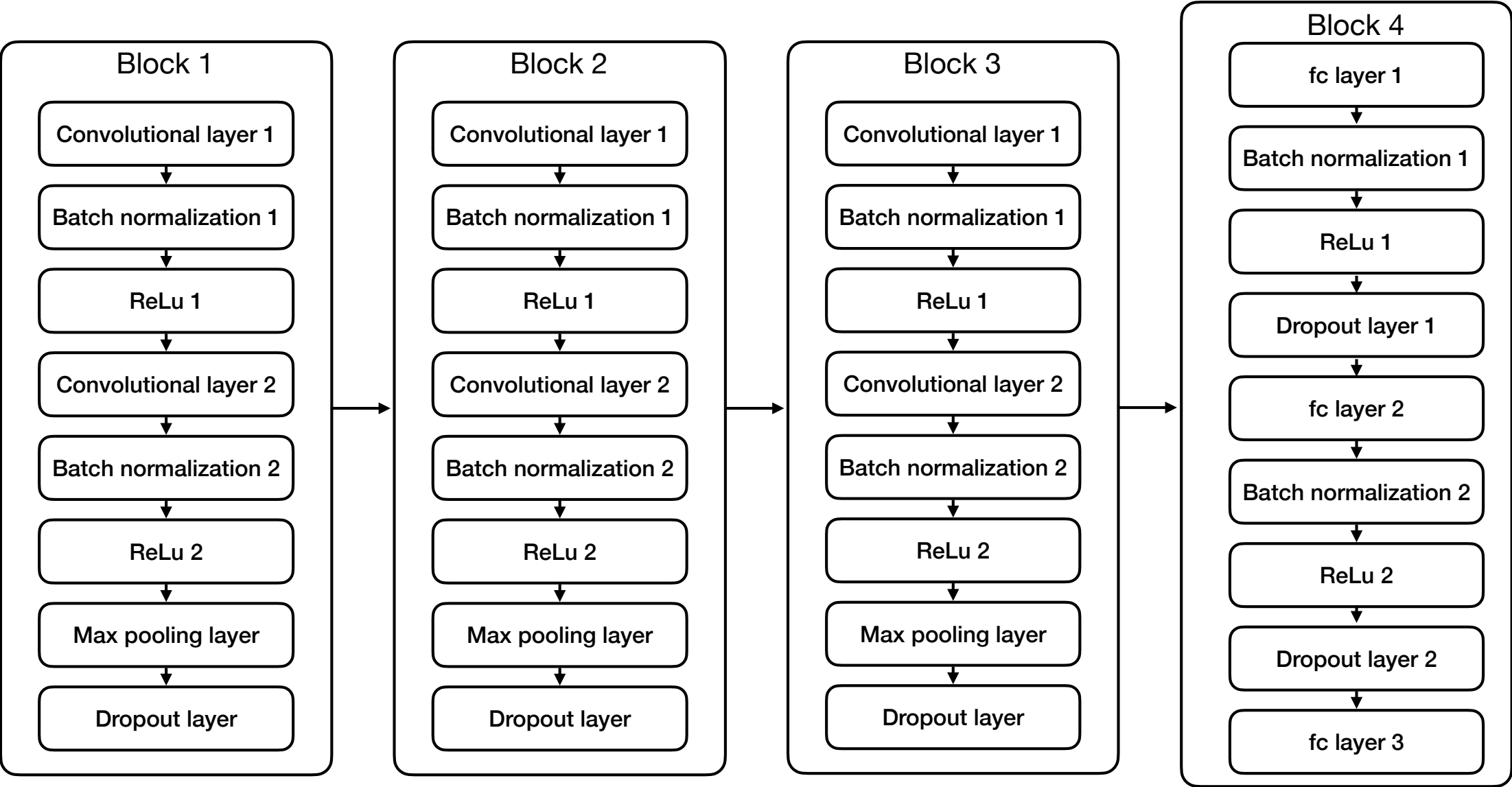
Wave-centered composites:



- Snapshots of each variable, tracking along the wave center. Temporal composites are the average of snapshots from the beginning of each wave's lifecycle till the end for non-devs and till becoming a hurricane for developers.
- 194 developers, 6055 weak non-devs, 320 strong non-devs (maximum $PV_{850} \geq 4.1 \times 10^{-5} s^{-1}$)
- Strong non-devs has comparable quantities with developers, the results are more robust.
- 80% as training set, 10% as validation set, 10% as testing set.
- The training set is balanced using Synthetic Minority Oversampling Technique (SMOTE).

ML model architectures and comparison

• Customized CNN architecture



• Pre-existing architectures:

Model name	Function
ResNet	resnet18
Wide ResNet	wide_resnet50_2
ResNeXt	resnext50_32x4d
EfficientNet	efficientnet_b0
VGG	vgg16
ViT	vit_b_16

Hyperparameters are optimized using Optuna, a Bayesian optimization framework that automatically searches better hyperparameters based on a Tree-structured Parzen Estimator (TPE). The validation f_{β} -score ($\beta = 0.75$) is used as optimization parameter. 100 trials for each model.

• Loss functions:

name	Function
Binary Cross Entropy (BCE) Weighted BCE	BCEWithLogitsLoss pos_weights= Strong non-dev/developer

• Optimizer:

name	Function
Adaptive Moment Estimation (Adam)	Adam
Adam with weight decay	AdamW
Stochastic Gradient Descent	SGD

• Other hyperparameters:

name	range/value
Random seed	42
Weight decay rate	0.0001
Scheduler	ReduceLROnPlateau
Learning rate	$[10^{-5}, 10^{-2}]$
Batch size	16, 32, 64

• Results

Model	Developer fbeta	Precision	Recall
Customized CNN	0.69	0.64	0.80
ResNet	0.70	0.68	0.72
Wide ResNet	0.77	0.82	0.69
ResNeXt	0.75	0.81	0.67
EfficientNet	0.77	0.92	0.59
VGG	0.75	0.83	0.64
ViT	0.71	0.71	0.64

Control experiments using AWS Autogluon multimodal function

Model: PyTorch Image Models (TIMM); Validation metric: Binary AUROC; Loss function: Cross Entropy

Combination	Strong non-dev f1	developer f1	macro avg	weighted avg
MSE_adv+MSE+PV	0.77	0.53	0.65	0.68
MSE_adv	0.76	0.33	0.55	0.60
MSE	0.77	0.00	0.38	0.48
PV	0.62	0.47	0.54	0.56
MSE_adv+MSE	0.82	0.54	0.68	0.71
MSE_adv+PV	0.77	0.49	0.63	0.66
MSE+PV	0.77	0.55	0.66	0.69

Conclusions

1. The performance of all models are close, with f_β -score (developer) ranging from 0.69 to 0.77. The precision has a higher deviations, ranging from 0.64 to 0.92. And the recall ranges from 0.59 to 0.8.
2. filtering out weaker ones is important for a robust result.
3. Adding more variables does not always improve the results, even they are physically related. The variables should be weighted according to their physical importance. A deep understanding of the physics is needed.

Future plans

- Create more samples using NWP models or using data augmentation methods, e.g. adding noise to existing samples.
- Study the time series instead of using composites.
- Better models (Nvidia Earth-2 AI stack) and adding more physical variables such as barotropical conversion.