Rapid emulation of a physics-based hydrology simulator with knowledge guided deep learning models

Andrew Bennett

Email: andrbenn@email.arizona.edu

THE UNIVERSITY OF ARIZONA

PRINCETON UNIVERSITY

HydroGEN

Climate change is hitting the Colorado River 'incredibly fast and incredibly hard'

The warming climate is intensifying drought, contributing to fires and drying out the river's headwaters, sending consequences cascading downstream.

Ian James, Arizona Republic

Published 7:37 AM MST Jan. 1, 2021 **| Updated 9:30 AM MST Jan. 8, 2021**

THE WALL STREET JOURNAL.

The Water Wars that Defined the American West Are Heading East

Urban growth and surge in irrigation fuel fight between Georgia and Florida; soybeans or oysters?

Groundwater is critical to understanding and simulating changing watersheds

Check for updates

PNAS PLUS

PNAS

Global models underestimate large decadal declining and rising water storage trends relative to GRACE satellite data

Bridget R. Scanlon[a,1], Zizhan Zhang[b], Himanshu Save[c], Alexander Y. Sun[a], Hannes Müller Schmied[d,e], Ludovicus P. H. van Beek[f], David N. Wiese[g], Yoshihide Wada[f,h], Di Long[i], Robert C. Reedy[a], Laurent Longuevergne[j], Petra Döll[d,e], and

LETTER

https://doi.org/10.1038/s41586-018-0424-4

Sensitivity of atmospheric CO_2 growth rate to observed changes in terrestrial water storage

Vincent Humphrey[1]*, Jakob Zscheischler[1], Philippe Ciais[2], Lukas Guc

LETTER

doi:10.1038/nature20780

Compensatory water effects link yearly global land CO_2 sink changes to temperature

Martin Jung[1], Markus Reichstein[1,2], Christopher R. Schwalm[3], Chris Huntingford[4], Stephen Sitch[5], Anders Ahlström[6,7]

HydroGEN

HydroGEN is a web-based platform that allows users to generate **seasonal forecasts** of watershed conditions **on demand**

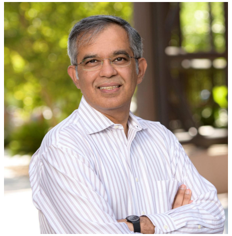
We are a very interdisciplinary team across academia, industry, and government agencies

- Hydrologists
- ML Experts
- Data Scientists
- Software Engineers
- User Experience Specialists

Laura Condon
Arizona

Reed Maxwell
Princeton

Peter Melchior
Princeton



Nirav Merchant
CyVERSE

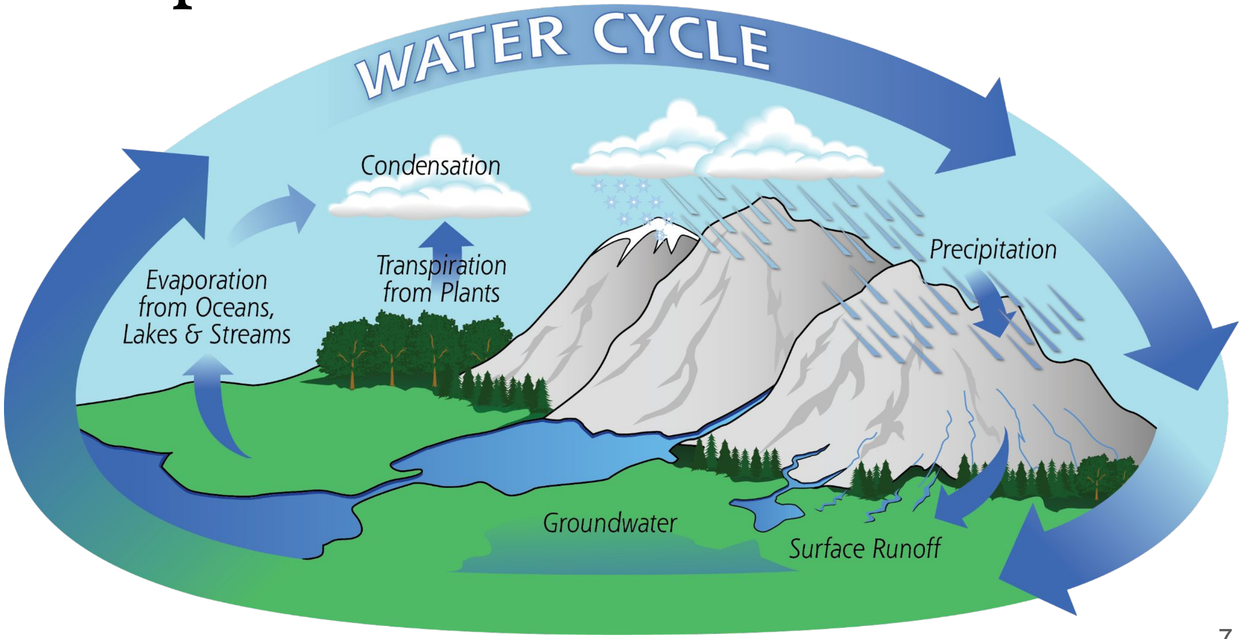
Elena Leonarduzzi
Princeton

Hoang Tran
PNNL

Ben Horowitz
Princeton

Lindsay Bearup
Bureau of Reclamation

Our goal is to jointly predict all major components of the terrestrial hydrologic cycle in a fully explicit spatiotemporal framework



https://gpm.nasa.gov/education/water-cycle

This leads to an inherent tension:

Spatially-distributed and multi-process implies process based models

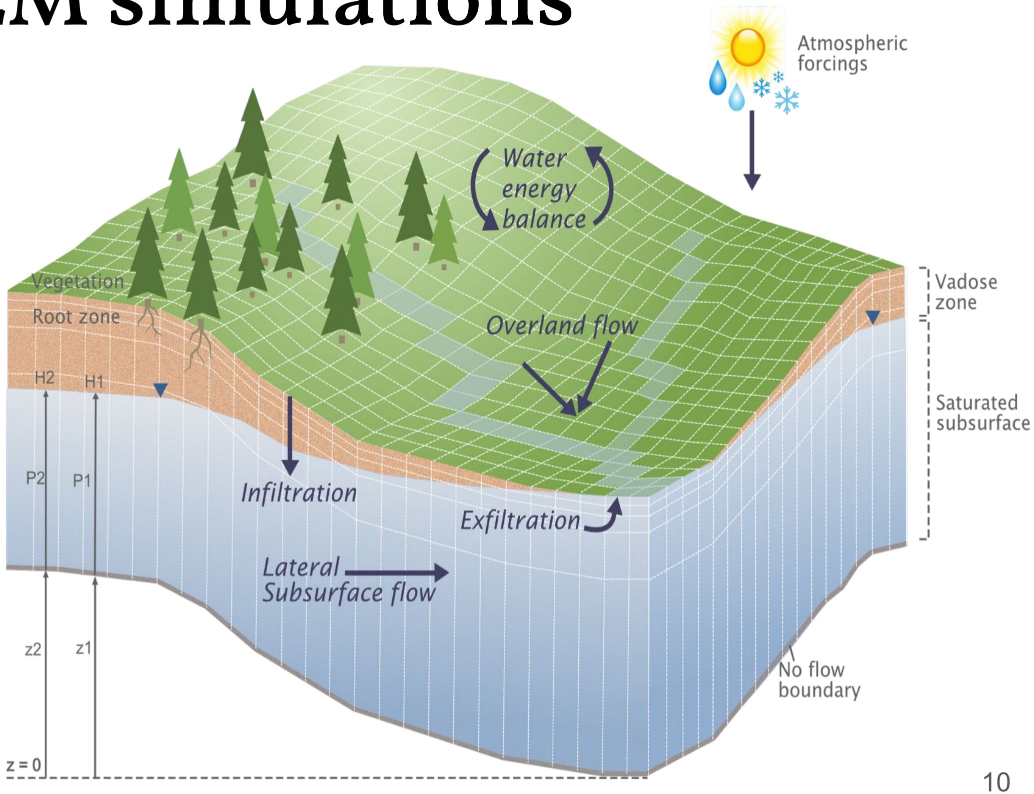
This leads to an inherent tension:

Spatially-distributed and multi-process implies process based models

But speed, flexibility, and predictive skill points toward data driven models

We developed an emulator based on coupled ParFlow-CLM simulations

- Unique national simulations that extend from the bedrock to the top of the treetops
- Solving physical equations for water and energy fluxes
- High resolution (1km)
- **Computationally expensive**

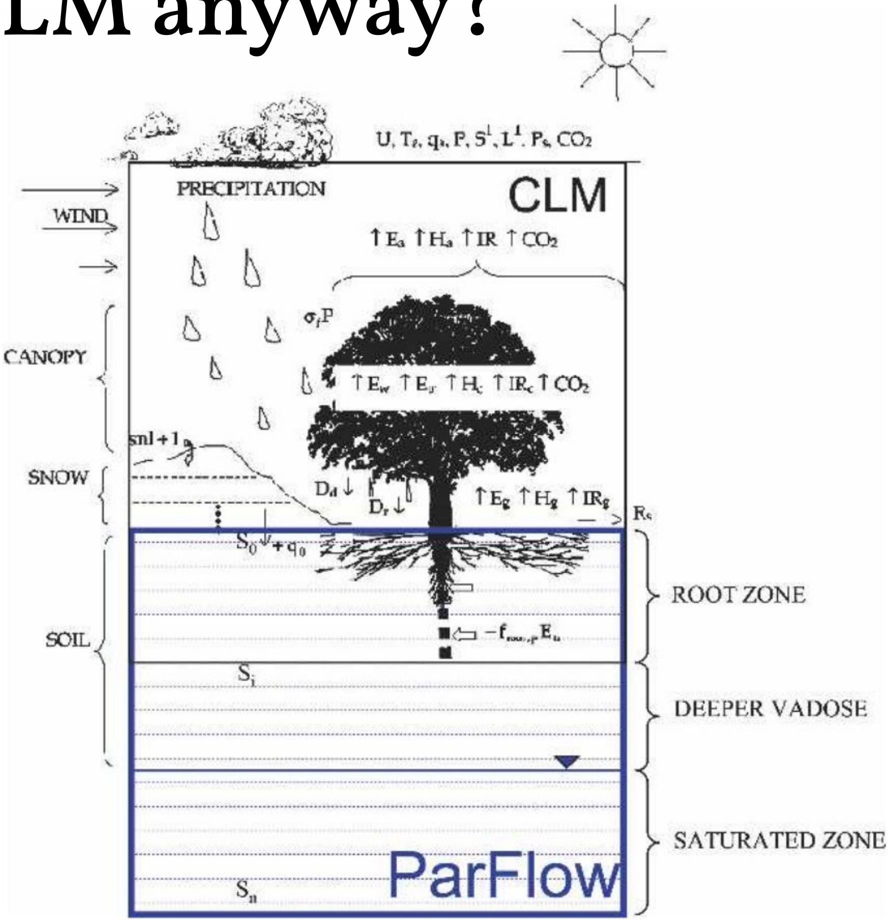


https://parflow.org/

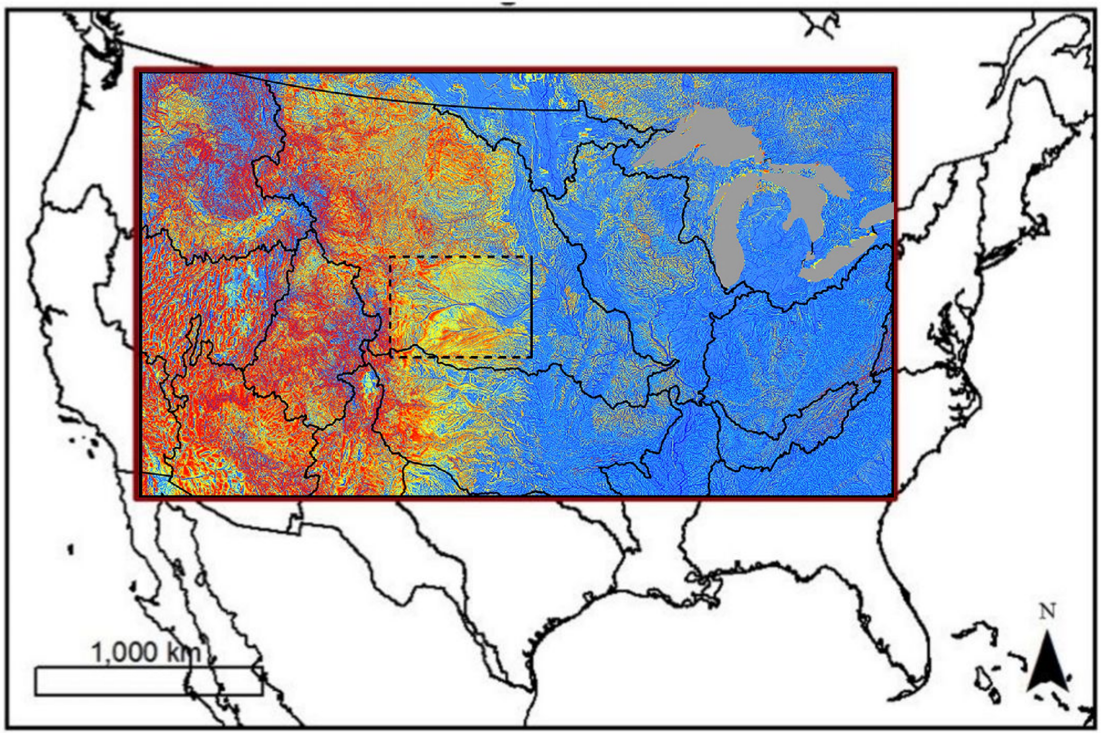
But what is ParFlow-CLM anyway?

CLM is a model of the land surface

ParFlow is a model of the subsurface

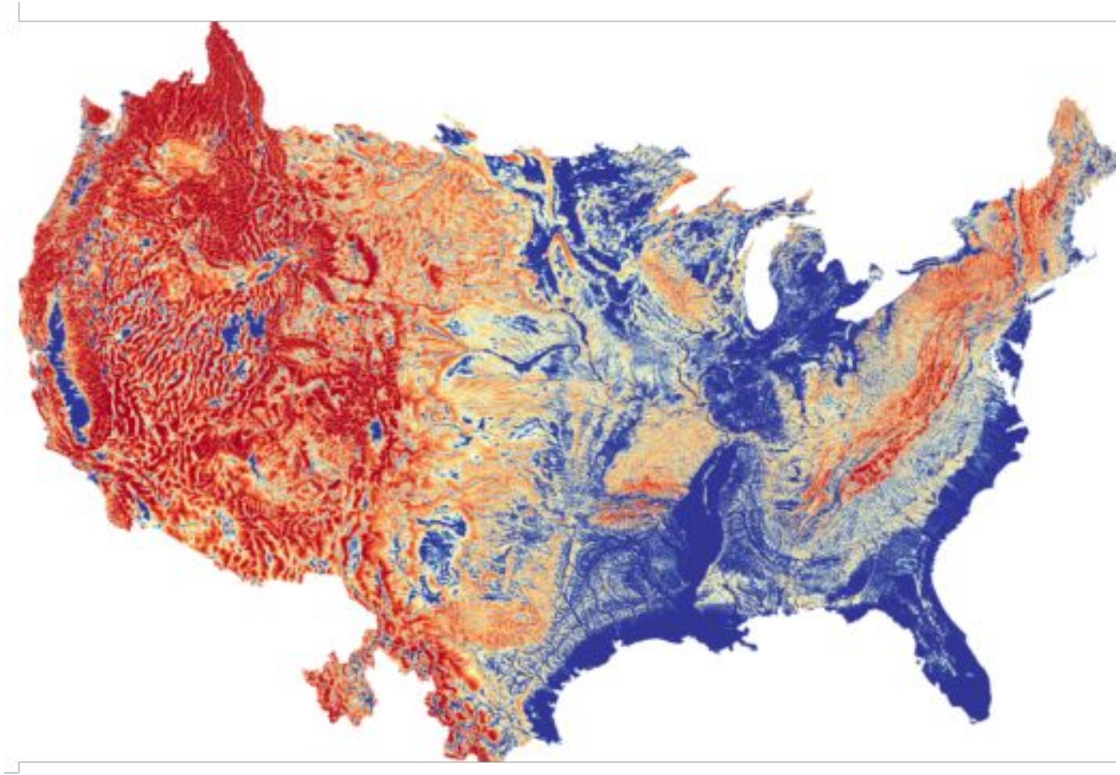


Our current model domain covers the majority of the Continental United States

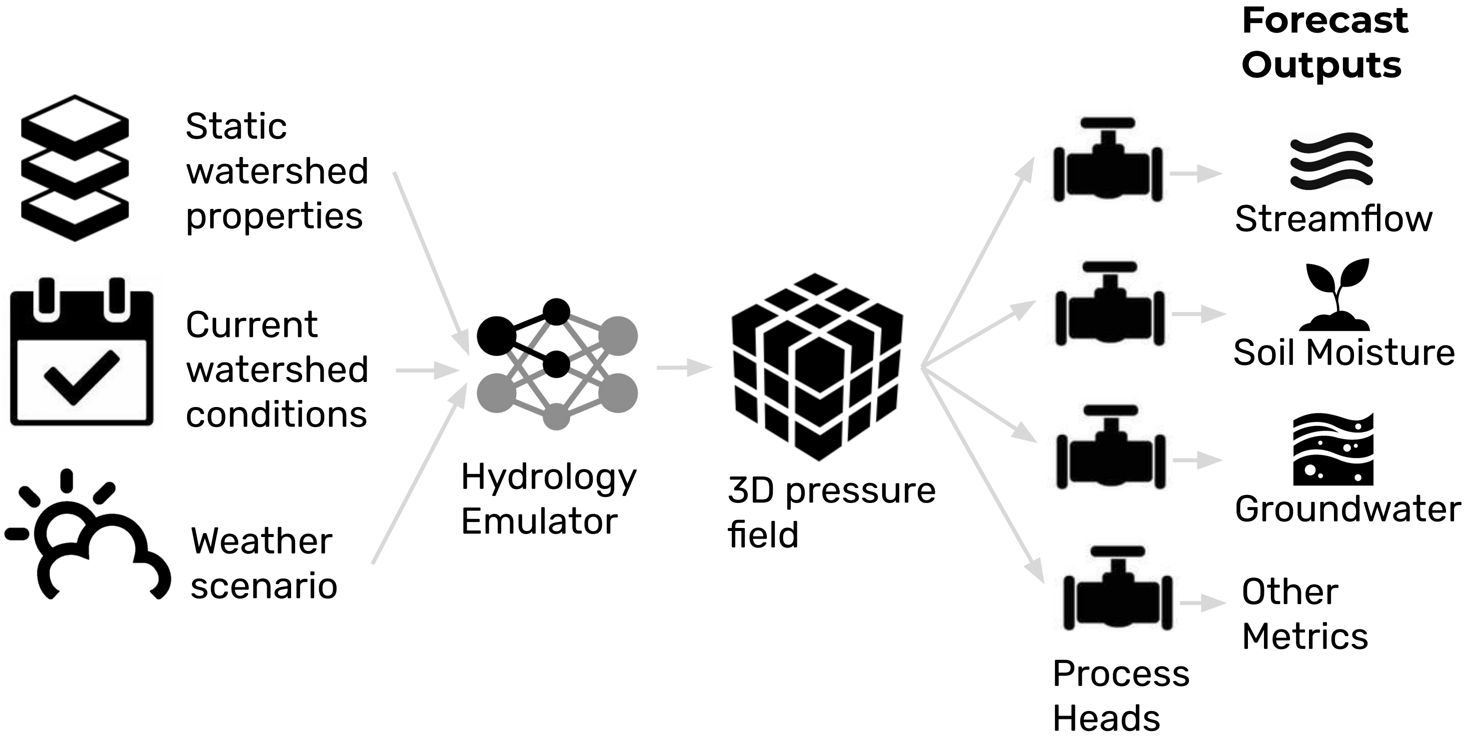


Maxwell, R. M., Condon, L. E., & Kollet, S. J. (2015). A high-resolution simulation of groundwater and surface water over most of the continental US with the integrated hydrologic model ParFlow v3. *Geoscientific Model Development*, *8*(3), 923–937. https://doi.org/10.5194/gmd-8-923-2015

We will be expanding to the full contiguous US soon!



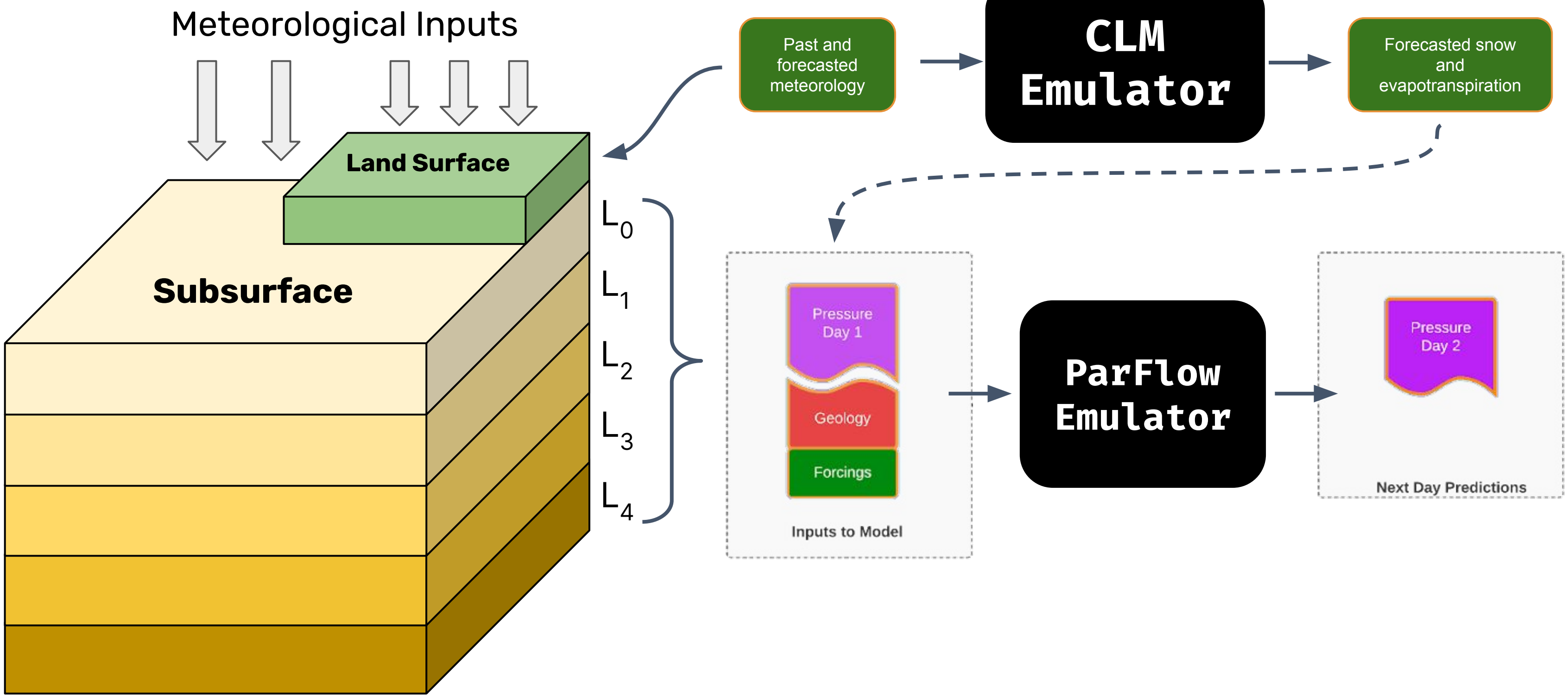
Our machine learning approach for forecast generation



Methods

$V = \frac{1}{3} \pi r^2 \cdot h$
h
r
30° 45° 60°
sin
cos
tan
$y = ax^2 + bx + c$
2a
2x
60°
30°
45°

The emulator approach



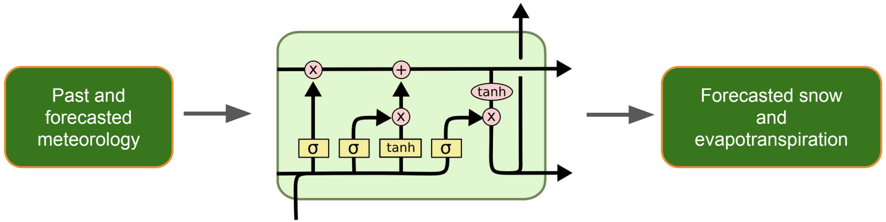
The land surface component is an LSTM

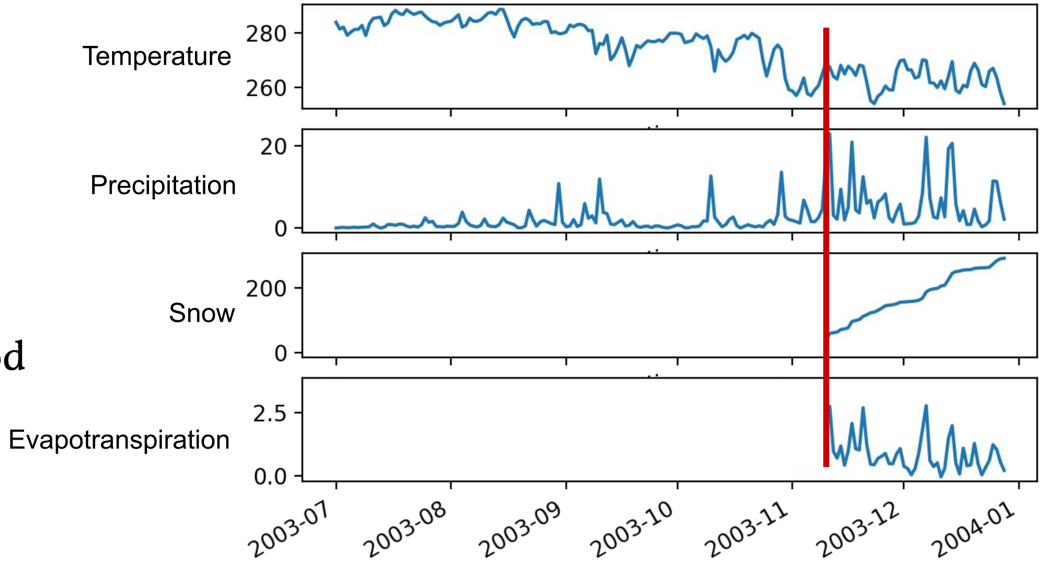
Pretty simple architecture, just a couple LSTM layers with about 256 nodes

Inputs (lookback 180 days) are:

- Max temp
- Min temp
- Precipitation
- Shortwave radiation
- Topographic index

Outputs are SWE and ET for last 60 (configurable) days of the input period





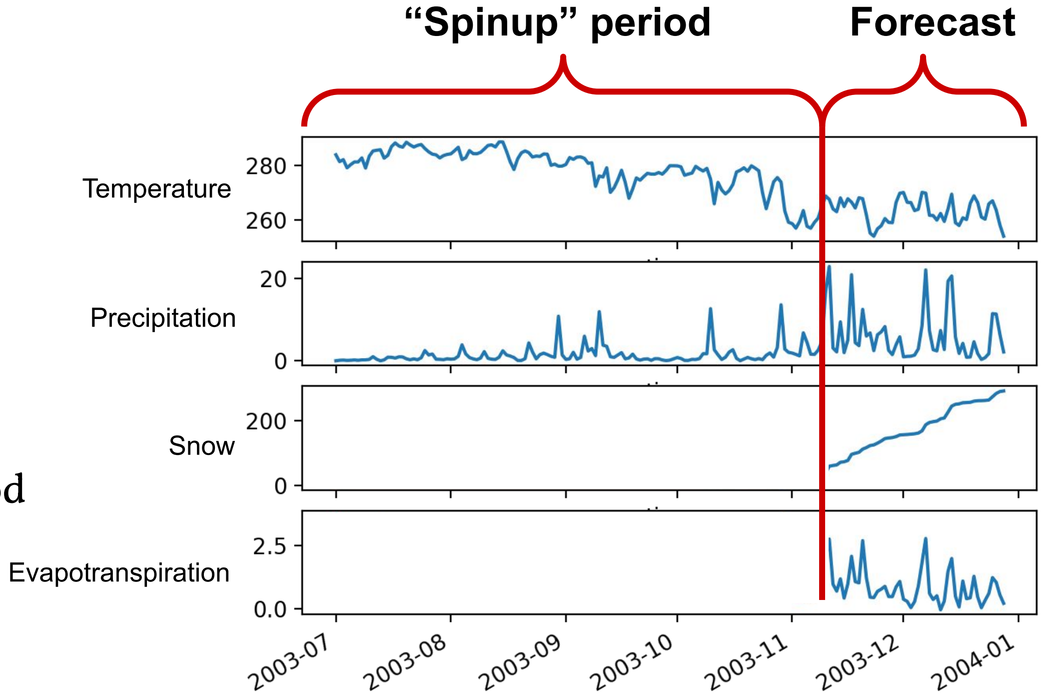
The land surface component is an LSTM

Pretty simple architecture, just a couple LSTM layers with about 256 nodes

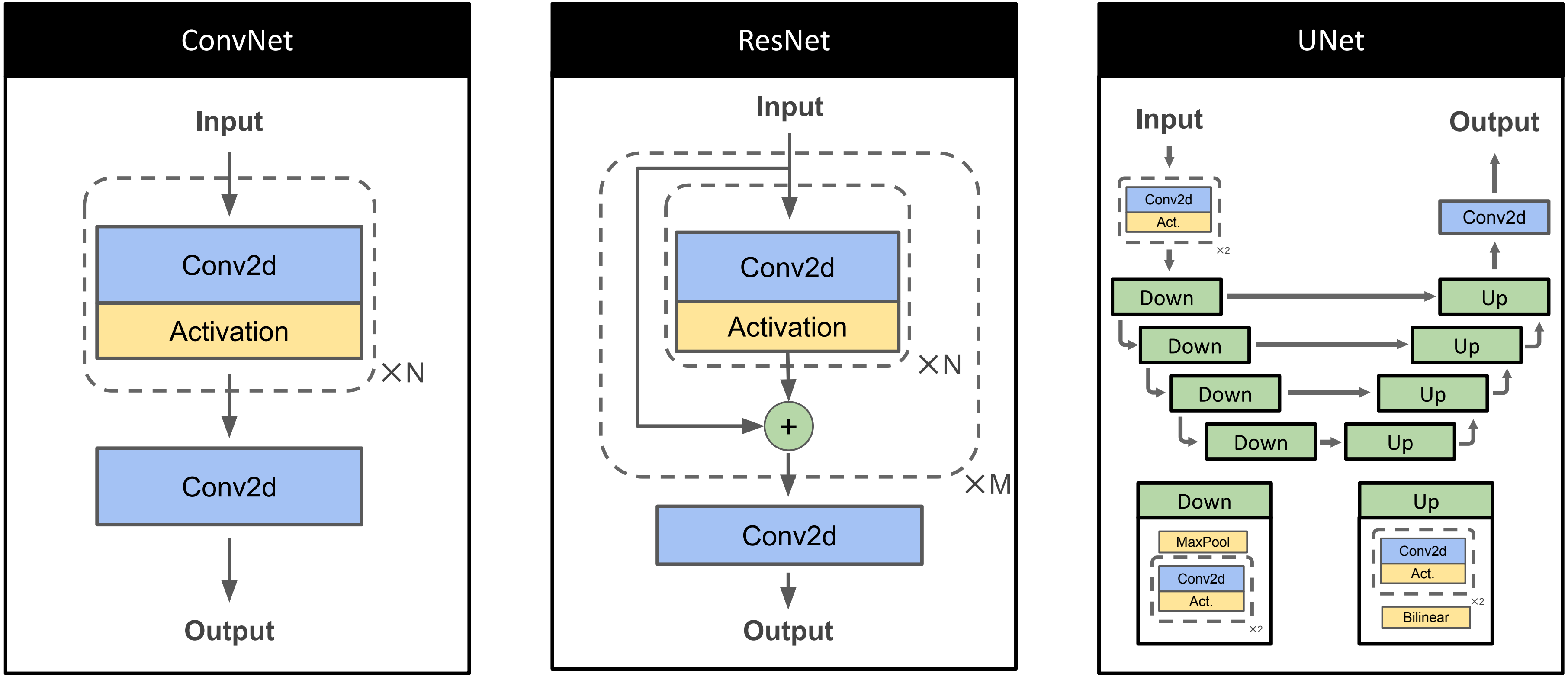
Inputs (lookback 180 days) are:

- Max temp
- Min temp
- Precipitation
- Shortwave radiation
- Topographic index

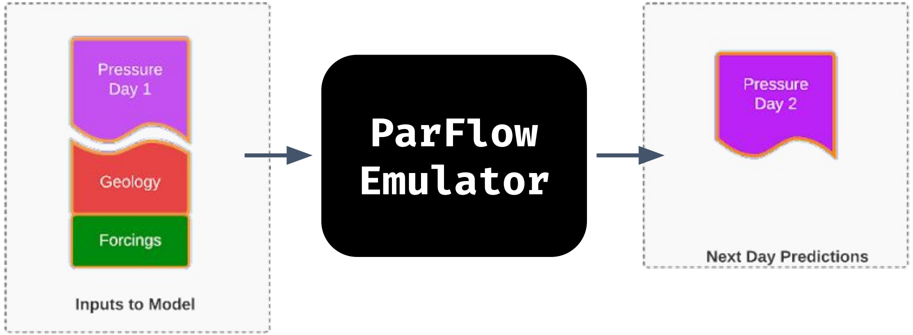
Outputs are SWE and ET for last 60 (configurable) days of the input period



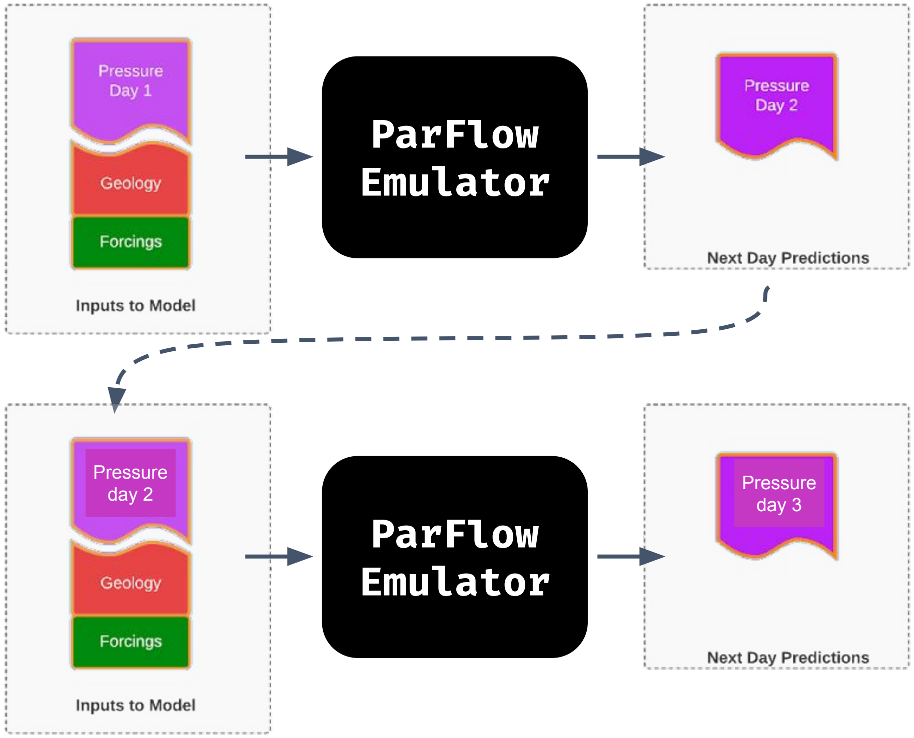
We have multiple architectures for the ParFlow emulator piece



To train the ParFlow emulator we unroll in time

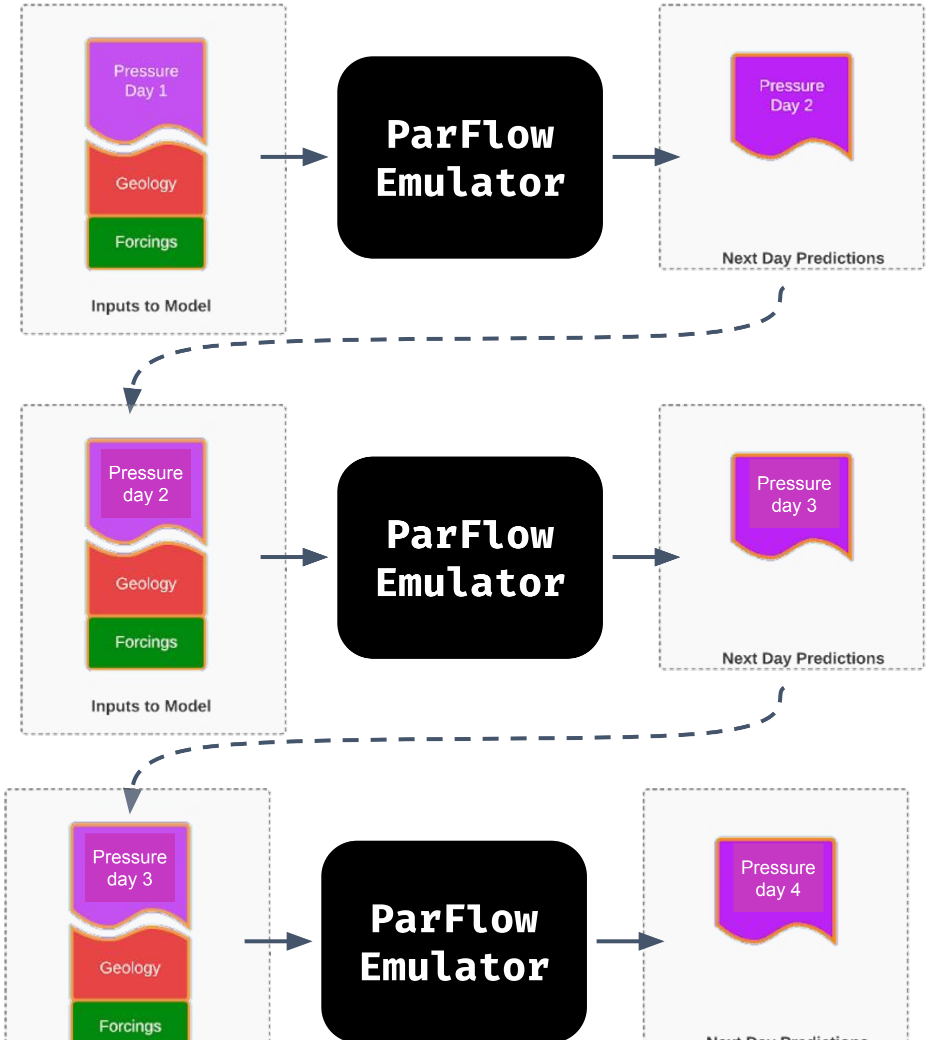


To train the ParFlow emulator we unroll in time

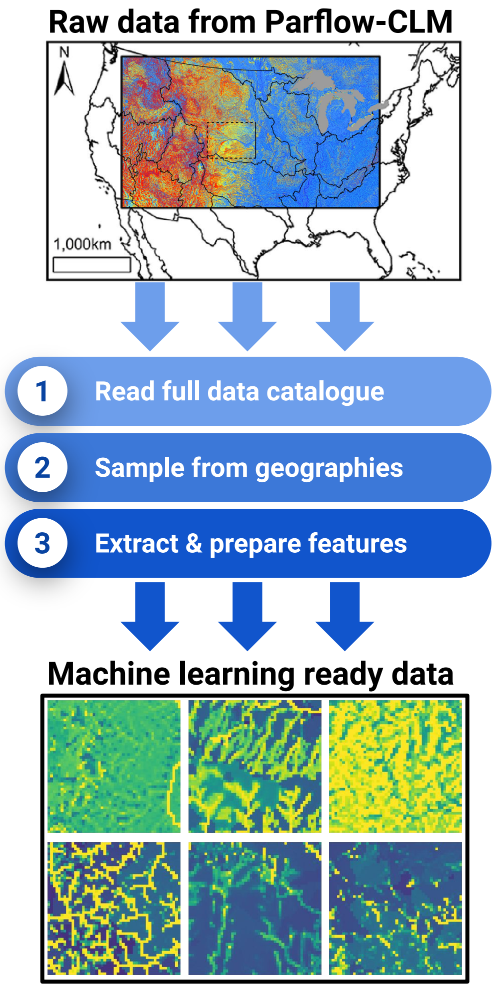


To train the ParFlow emulator we unroll in time

During training we increase the rollout, usually to ~30 days

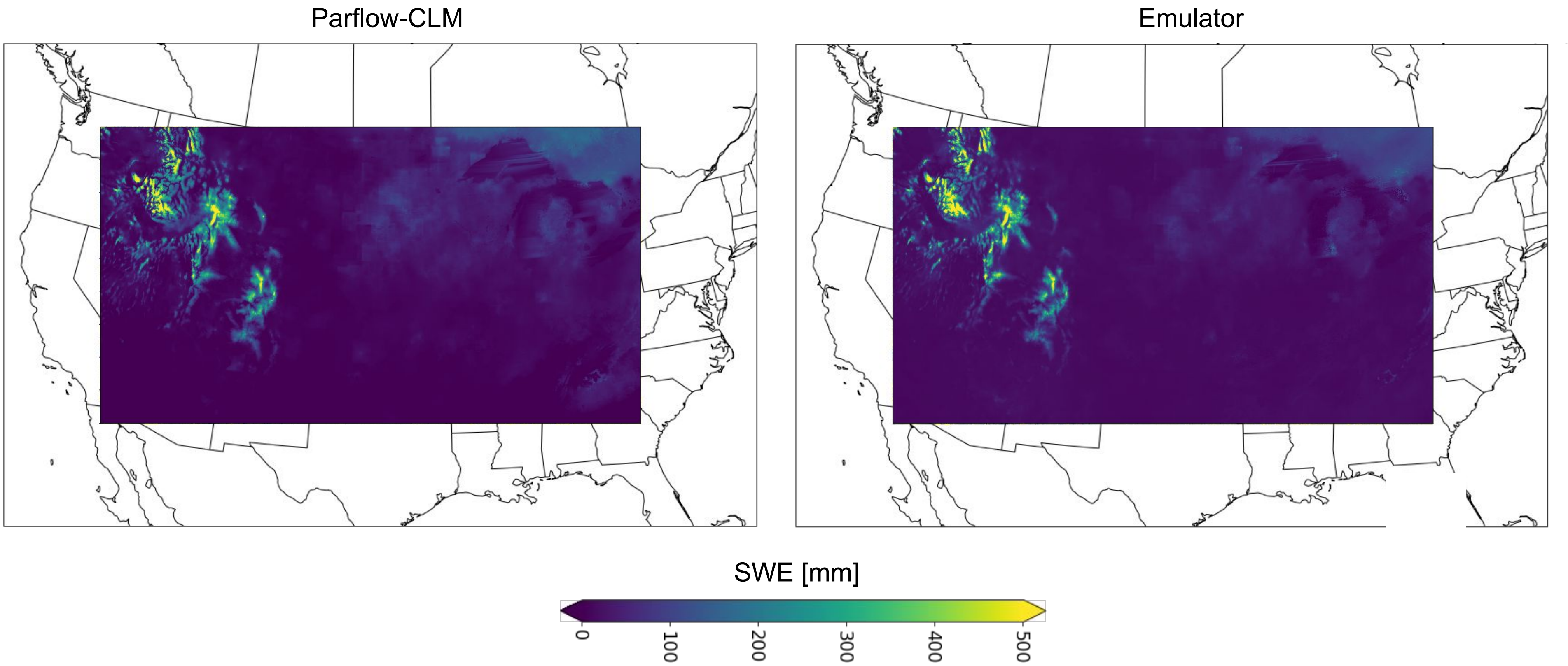


The training data is sampled patch-wise from the full domain

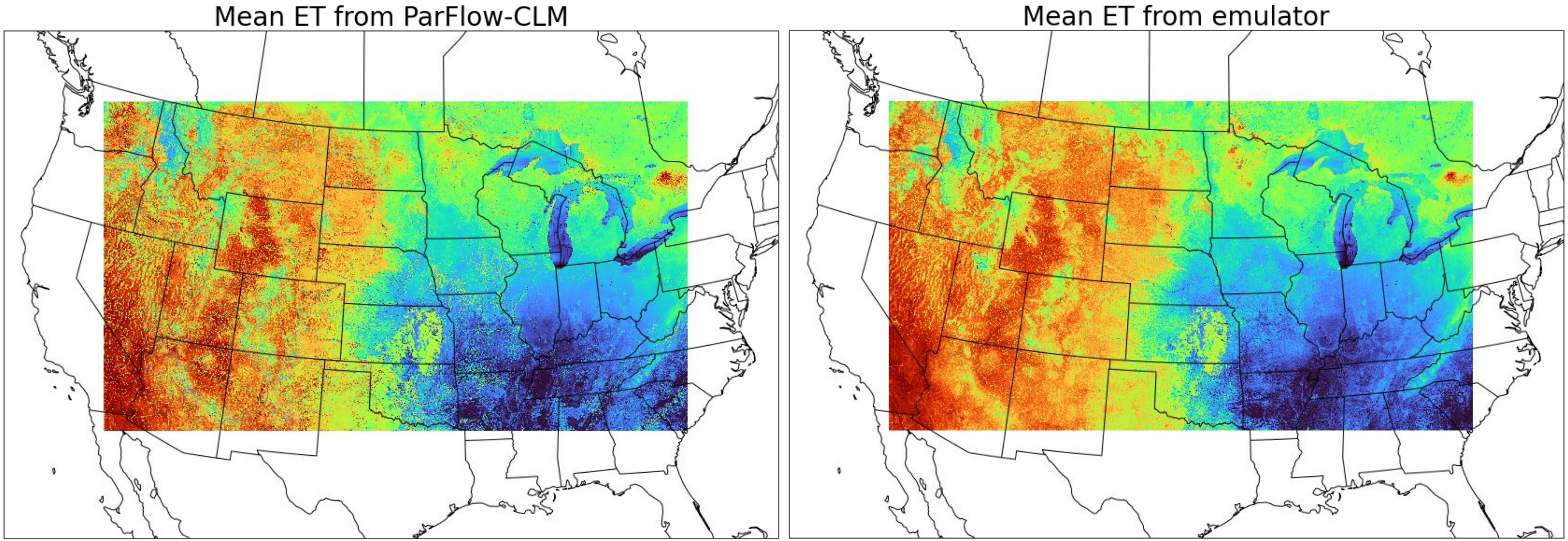


Some brief results

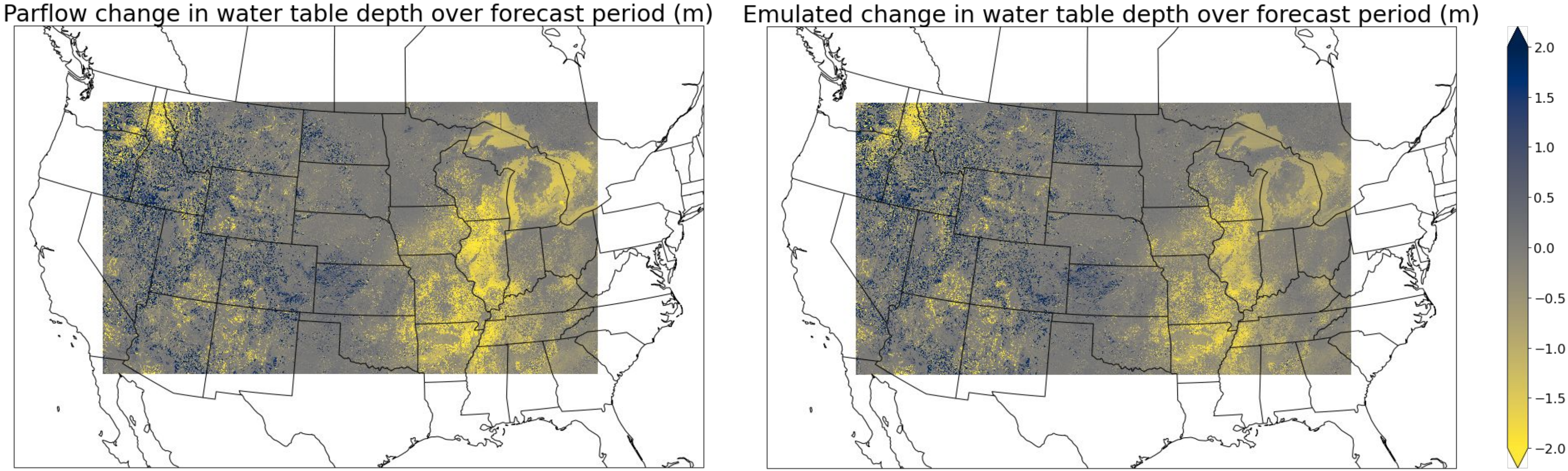
We’re able to match snow patterns generally



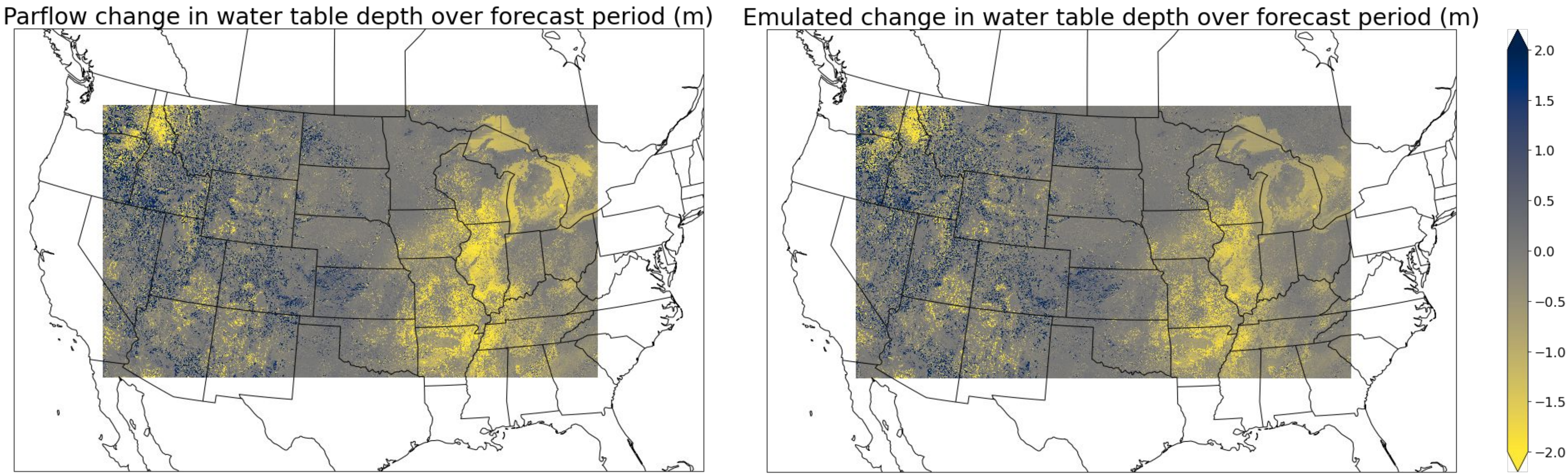
Same for ET, largely



Emulator shows promising performance across the Continental US (CONUS)

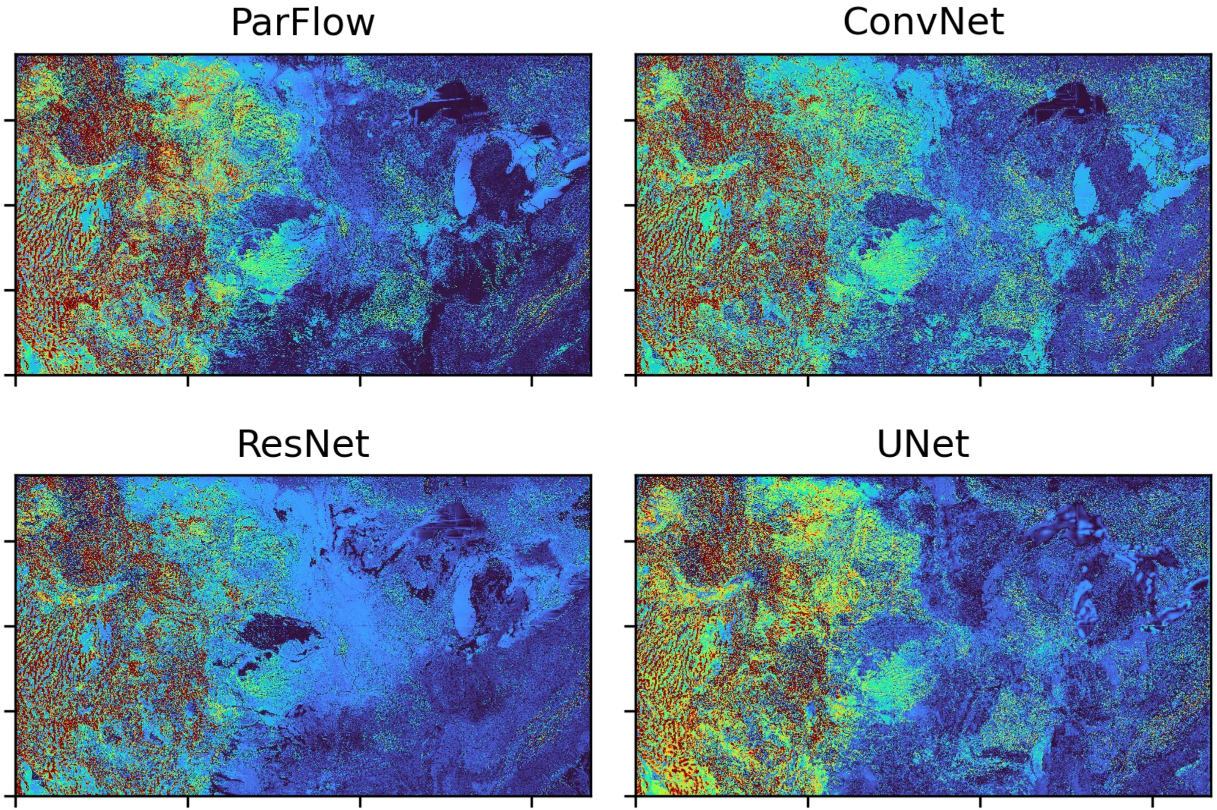


We get good performance in the subsurface generally too



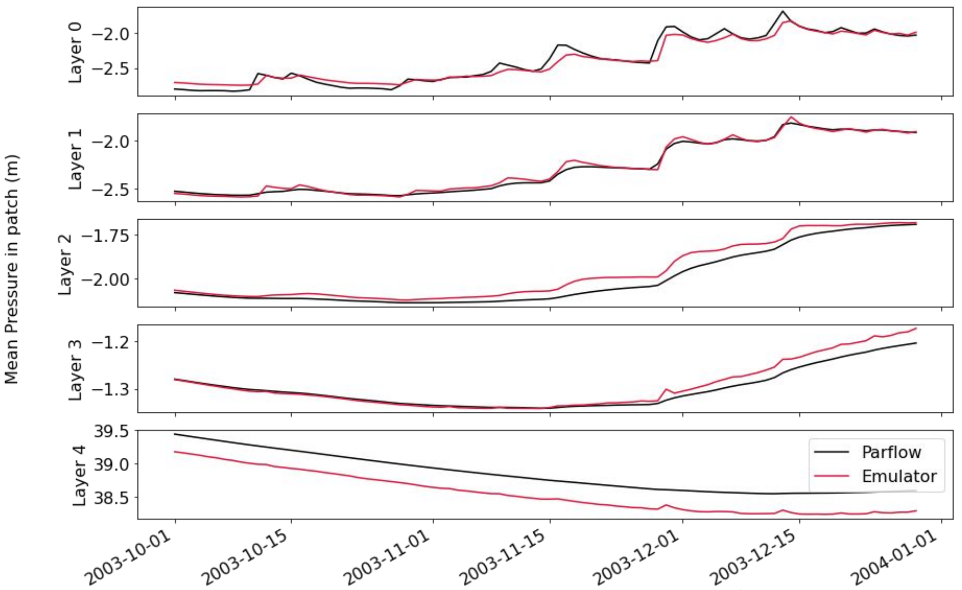
- 90 day forecast takes ~10 minutes on 1 GPU
- >50x speedup vs Parflow on 1,000 cores
- Saturation, water table depth and streamflow calculated by process heads off the simulated pressure field

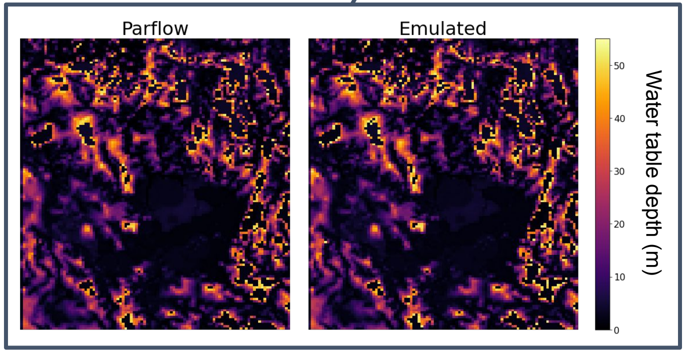
Model architectures have different performance characteristics, but generally all capture broad patterns



90 day rollouts match dynamics well too

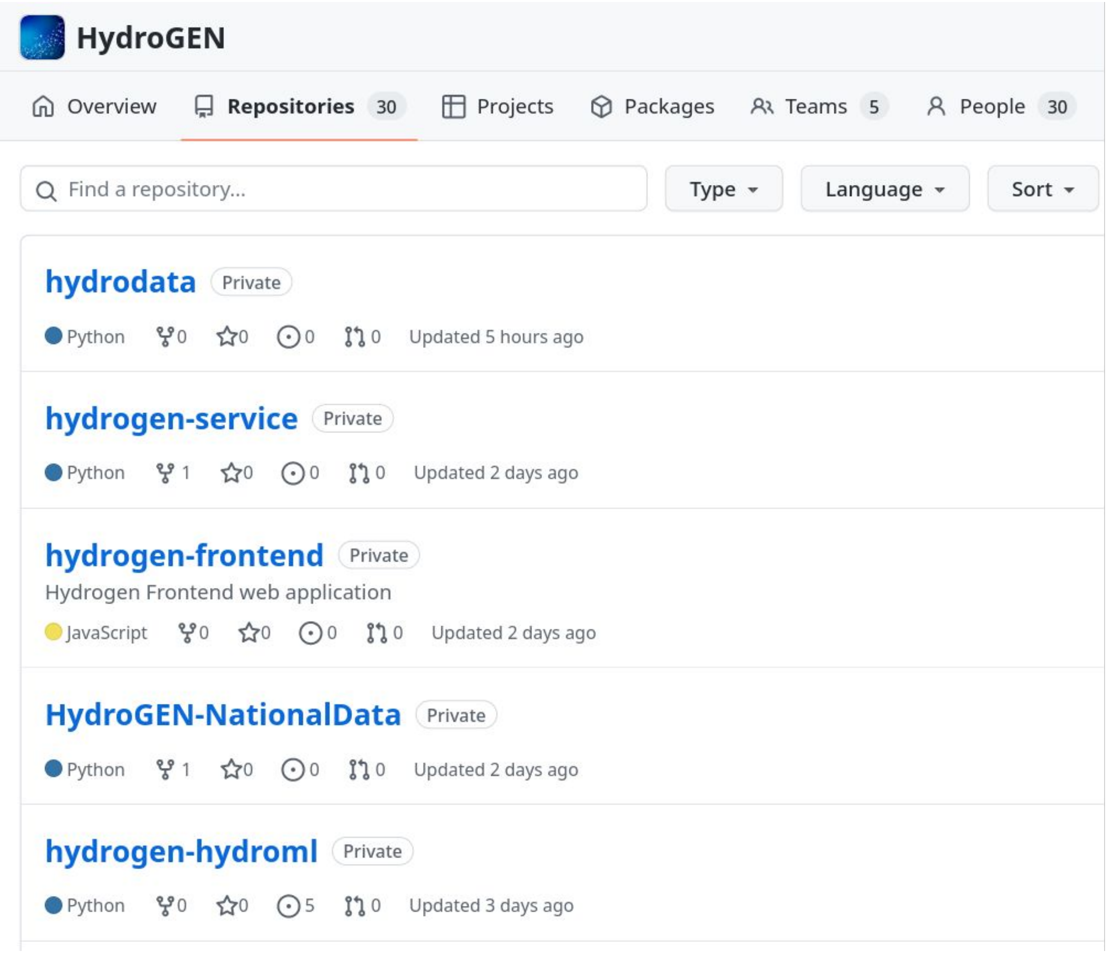
- Here we show 90 timeseries for all 5 layers of the pressure
- Showing for a location in eastern Oregon state marked by red star on map
- Again, patterns are captured well, but bottom layer has a constant bias
- Zooming in we see spatial patterns of water table depth are very well captured





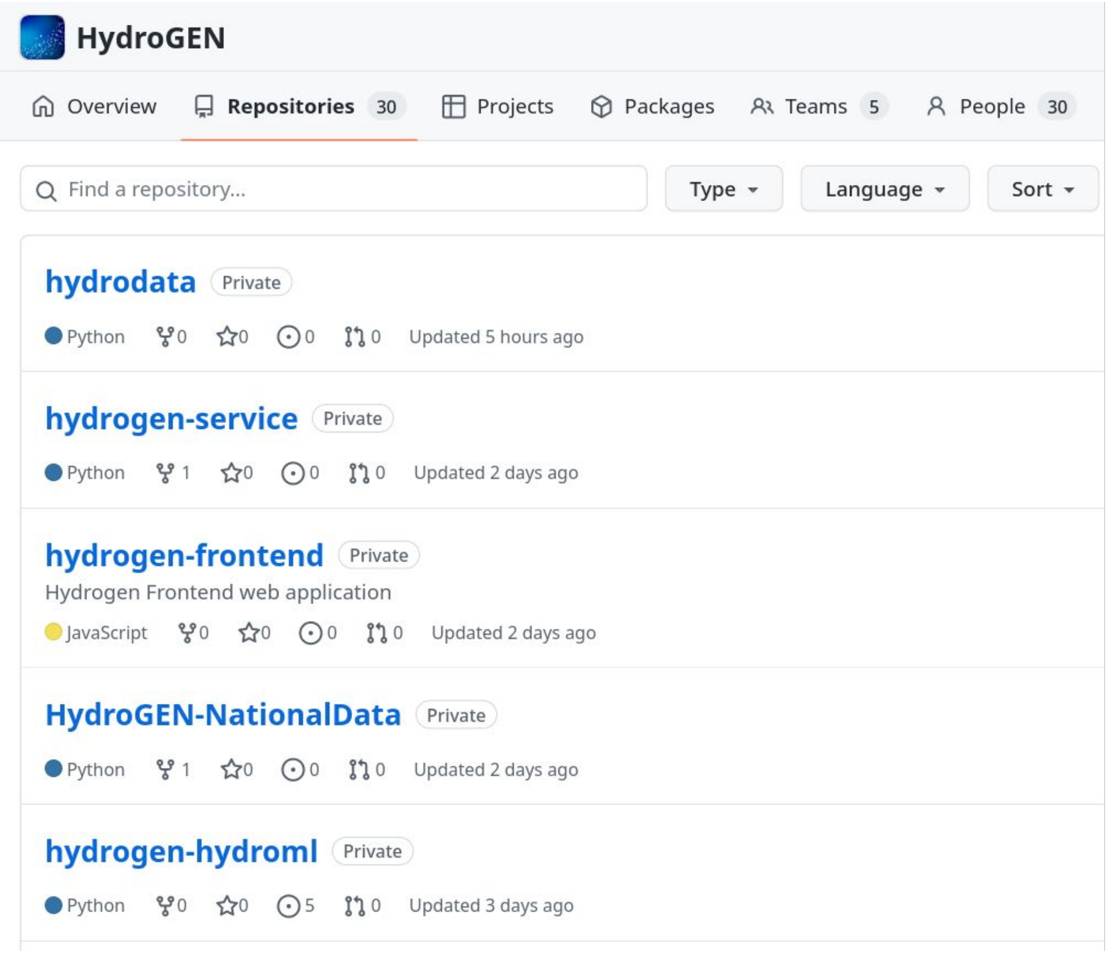
The messy bits

We are actually deploying these models in a webapp



We are actually deploying these models in a webapp

It turns out MLOps is not an easy task

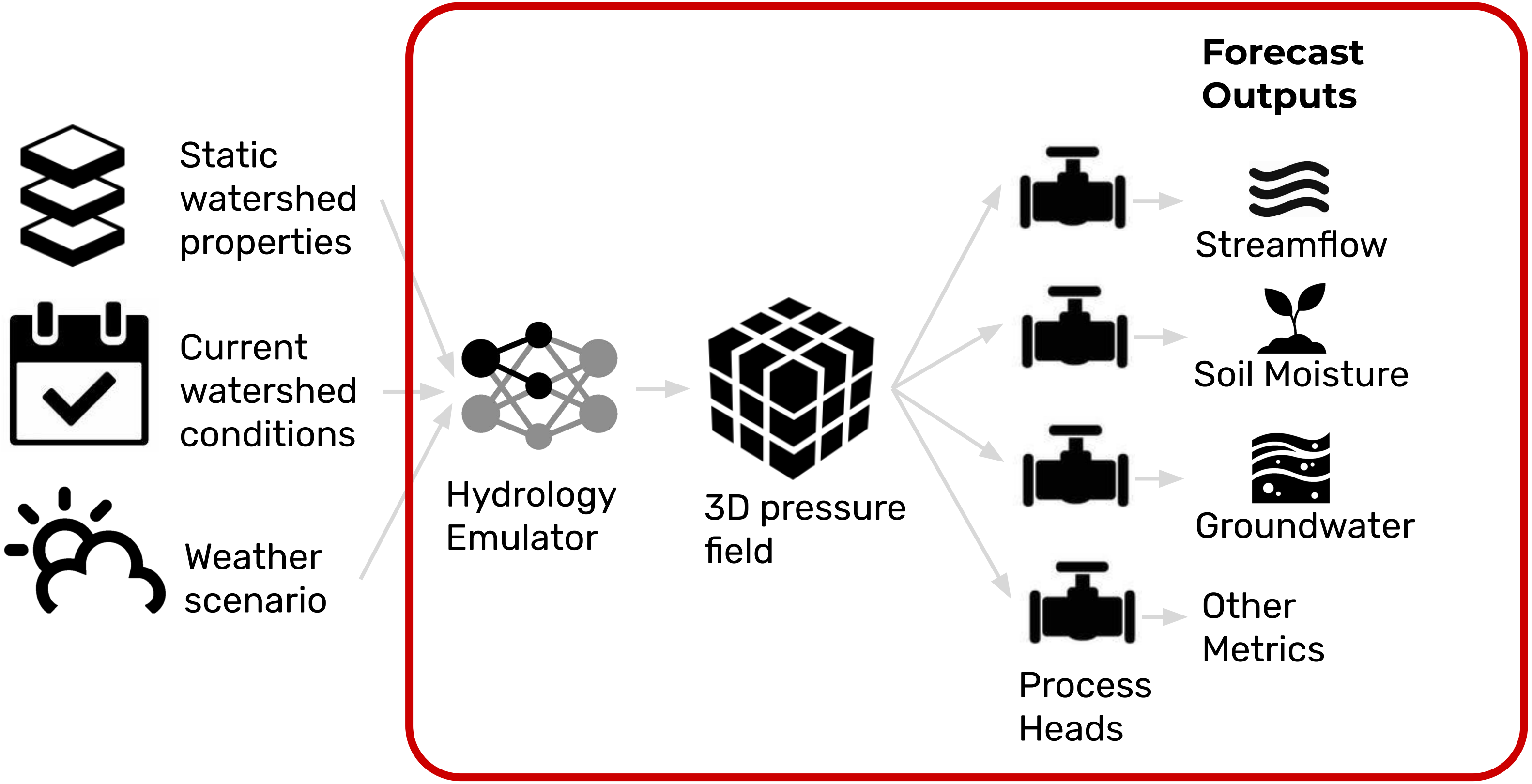


MLFlow has been key for tracking and preparing models for deployment

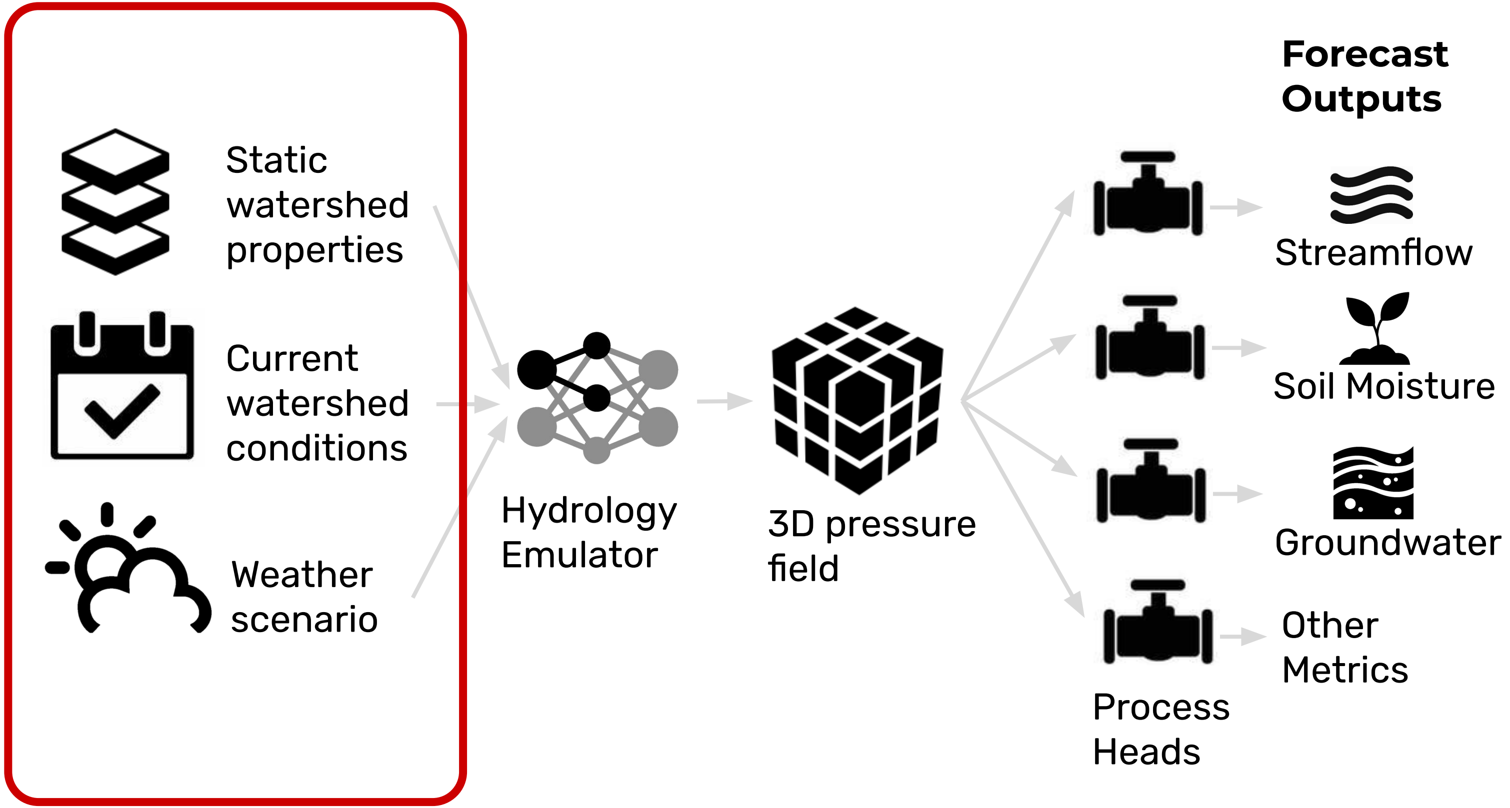
MLFlow has been key for tracking and preparing models for deployment

But still at the end of the day, we end up with some shims for the “model zoo”

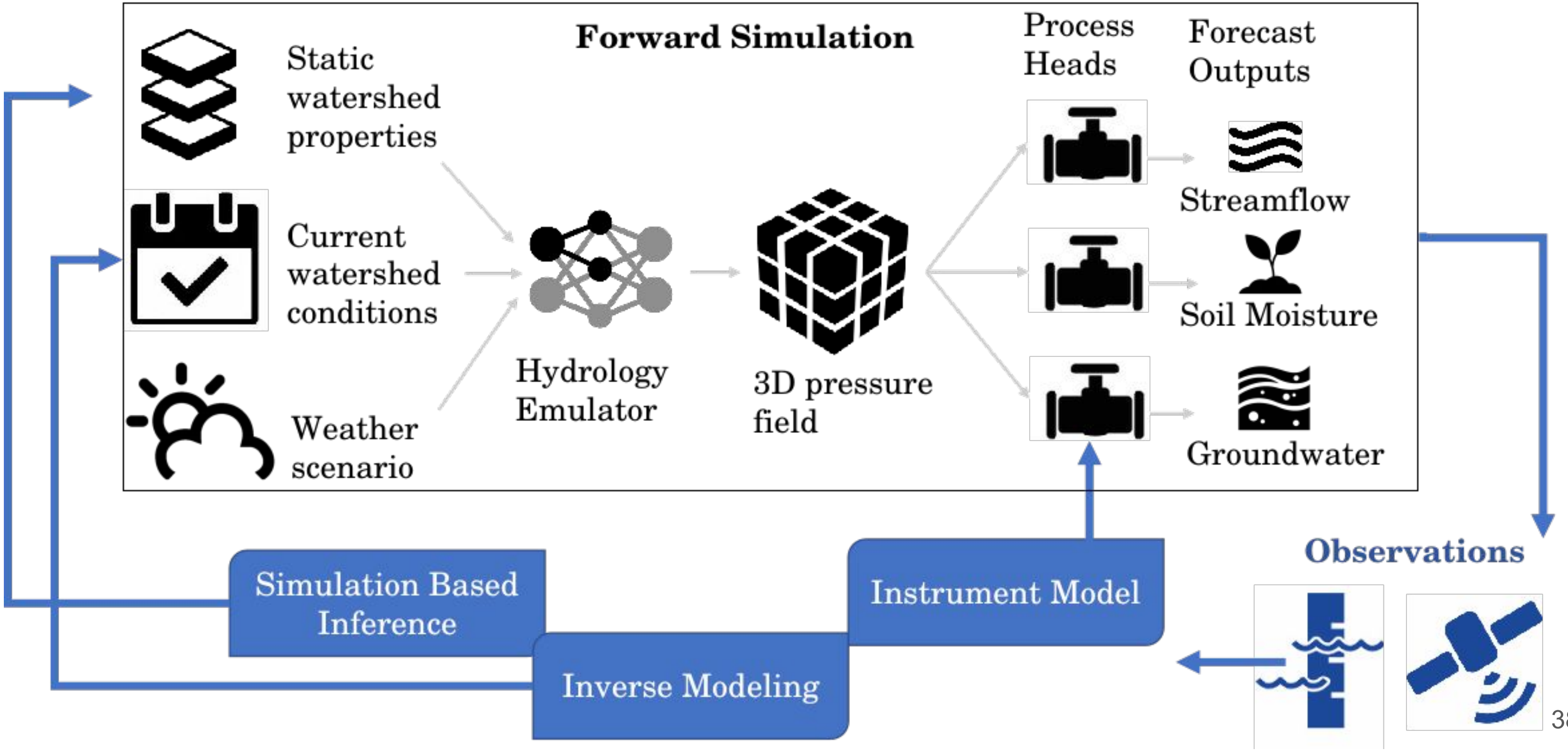
And that's not even the full system



Each of these pieces need to be automated too!



To do this we are bringing in a diverse sets of observations



Still tons of work to do…

- Continue to improve emulator by training on more data, using data augmentation, more training steps, etc
- Perform more comprehensive selection of input data for surface module
- Train directly on derived processes such as water table depth and streamflow
- SBI and inverse modeling to get initial state and parameter values to be more predictive of observations

Still tons of work to do...

- Continue to improve emulator by training on more data, using data augmentation, more training steps, etc
- Perform more comprehensive selection of input data for surface module
- Train directly on derived processes such as water table depth and streamflow
- SBI and inverse modeling to get initial state and parameter values to be more predictive of observations

Thanks for listening! Questions?

Email: andrbenn@email.arizona.edu

Twitter: @arbennett_

Time permitting, a demo!

https://hydro-dev.princeton.edu/