Experimenting with machine learning hindcast emulators to help tune CAM (a.k.a wherein I attempt "scikit-learn for dummies")

Colin Zarzycki (@weatherczar)
+ others!

[MomentumCPT]

PennState

- Two primary core code-level improvements…
  - Directly prognosing momentum flux evolution in CLUBB

$$\frac{\partial \overline{u_h'w'}}{\partial t} = \underbrace{-\overline{w}\frac{\partial \overline{u_h'w'}}{\partial z}}_{ma} \underbrace{-\frac{1}{\rho_s}\frac{\partial \rho_s \overline{w'^2u_h'}}{\partial z}}_{ta} \underbrace{-\overline{w'^2}\frac{\partial \overline{u_h}}{\partial z}}_{tp} \underbrace{-\overline{u_h'w'}\frac{\partial \overline{w}}{\partial z}}_{ac} \underbrace{+\frac{g}{\theta_{vs}}\overline{u_h'\theta_v'}}_{bp}$$

$$\underbrace{-\frac{C_6}{\tau}\overline{u_h'w'}}_{pr1} \underbrace{+C_7\overline{u_h'w'}\frac{\partial \overline{w}}{\partial z}}_{pr2} \underbrace{-C_7\frac{g}{\theta_{vs}}\overline{u_h'\theta_v'}}_{pr3} \underbrace{+C_7\overline{w'^2}\frac{\partial \overline{u_h}}{\partial z}}_{pr4}$$

$$\underbrace{+\frac{\partial}{\partial z}\left[(K_{w6}+\nu_6)\frac{\partial}{\partial z}\overline{u_h'w'}\right]}_{dp1}$$

Larson (2020), Guo et al., (2021), Nardi et al., (2022)

  - Defining a "regime-specific" eddy timescale formulation

$$L_{scale} = \tau\sqrt{TKE}$$

$$\frac{1}{\tau} = \underbrace{\frac{C_{\tau,back}}{\tau_{ref}}}_{1} \underbrace{+C_{\tau,sfc}\left(\frac{u^*}{K}\right)\left(\frac{1}{z-z_s+z_{dis}}\right)}_{2} \underbrace{+C_{\tau,shear}\left(\left(\frac{\partial u}{\partial z}\right)^2+\left(\frac{\partial v}{\partial z}\right)^2\right)^{\frac{1}{2}}}_{3} \underbrace{+C_{\tau,N}\sqrt{\max(N^2,0)}}_{4}$$

[MomentumCPT]

PennState

Parameter	EAM-def	EAM-taus
C1 = C1b	1.335	–
C14	1	–
C2rt = C2thl	1.75	–
C2rtthl	2.275	–
C6rt = C6thl	4	–
C6rtb = C6thlb	6	–
C6rtc = C6thlc	1	–
C6rt_Lscale0	14	–
C6thl_Lscale0	14	–
wpxp_L_thresh	60	–
C8	4.3	0.5
C11	0.8	0.5
C11b	0.35	0.5
C11c	0.5	–
gamma_coef	0.32	0.3
gamma_coefb	0.32	0.3
beta	1	2
C_{irsfc}	–	0.3
$C_{irshear}$	–	0.15
$C_{irbkgnd}$	–	1.5
C_{irN}	–	0.65
$C_{irN,clr}$	–	2.0
z_s	–	300 m
$C_{irwpxpRi}$	–	3
$C_{irxp2Ri}$	–	1
$z_{displace}$	–	10 m
N^2_{thresh}	–	3.3E–4

Guo et al., 2021

Trade ~10 parameters, **no added DoFs to CLUBB's tunable parameter set**

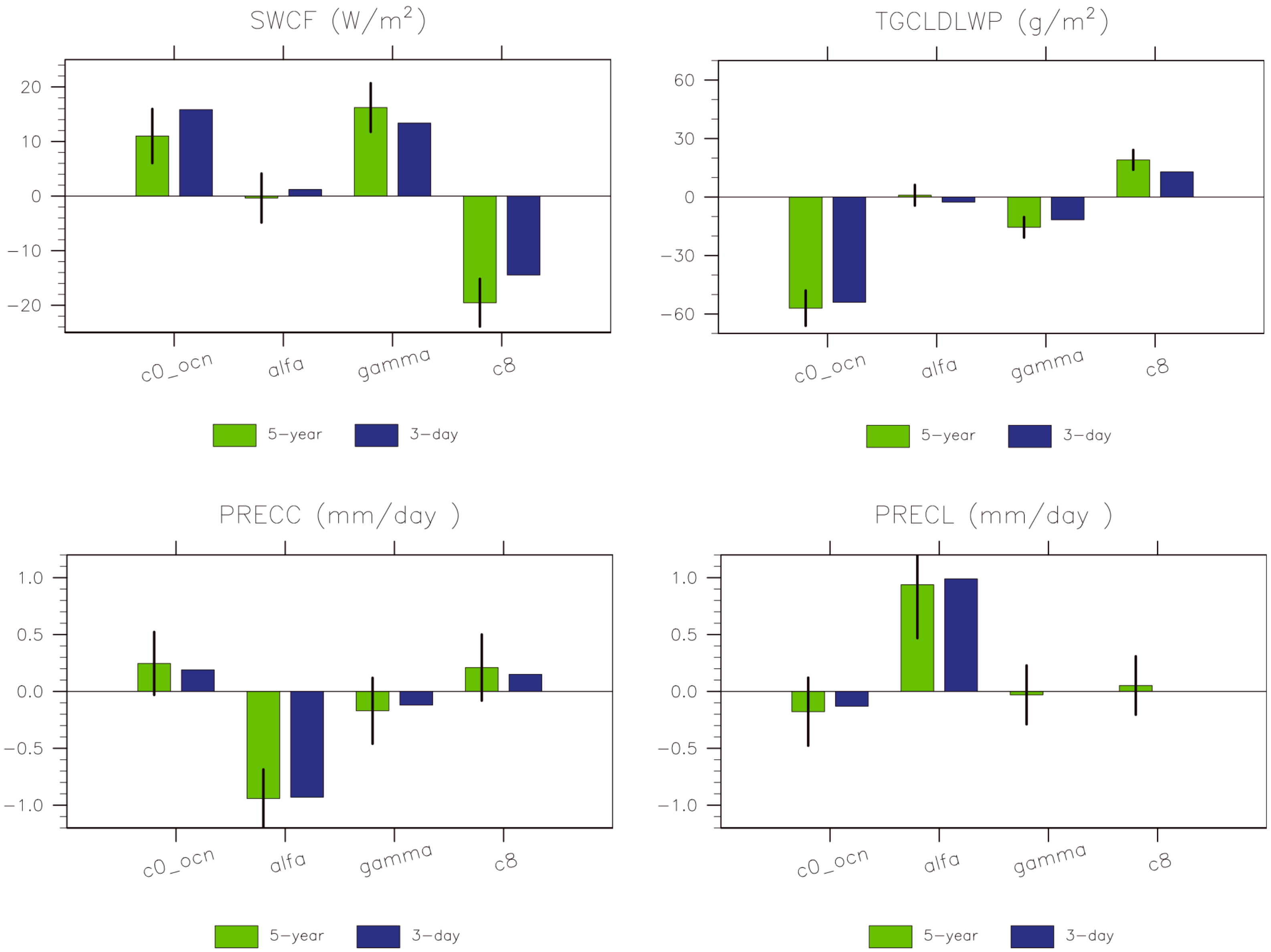
More "control" over momentum fluxes -> more <u>interpretable</u> from a process-oriented standpoint

Using short-term simulations to predict long-term response

PennState

3-day run

5-year run

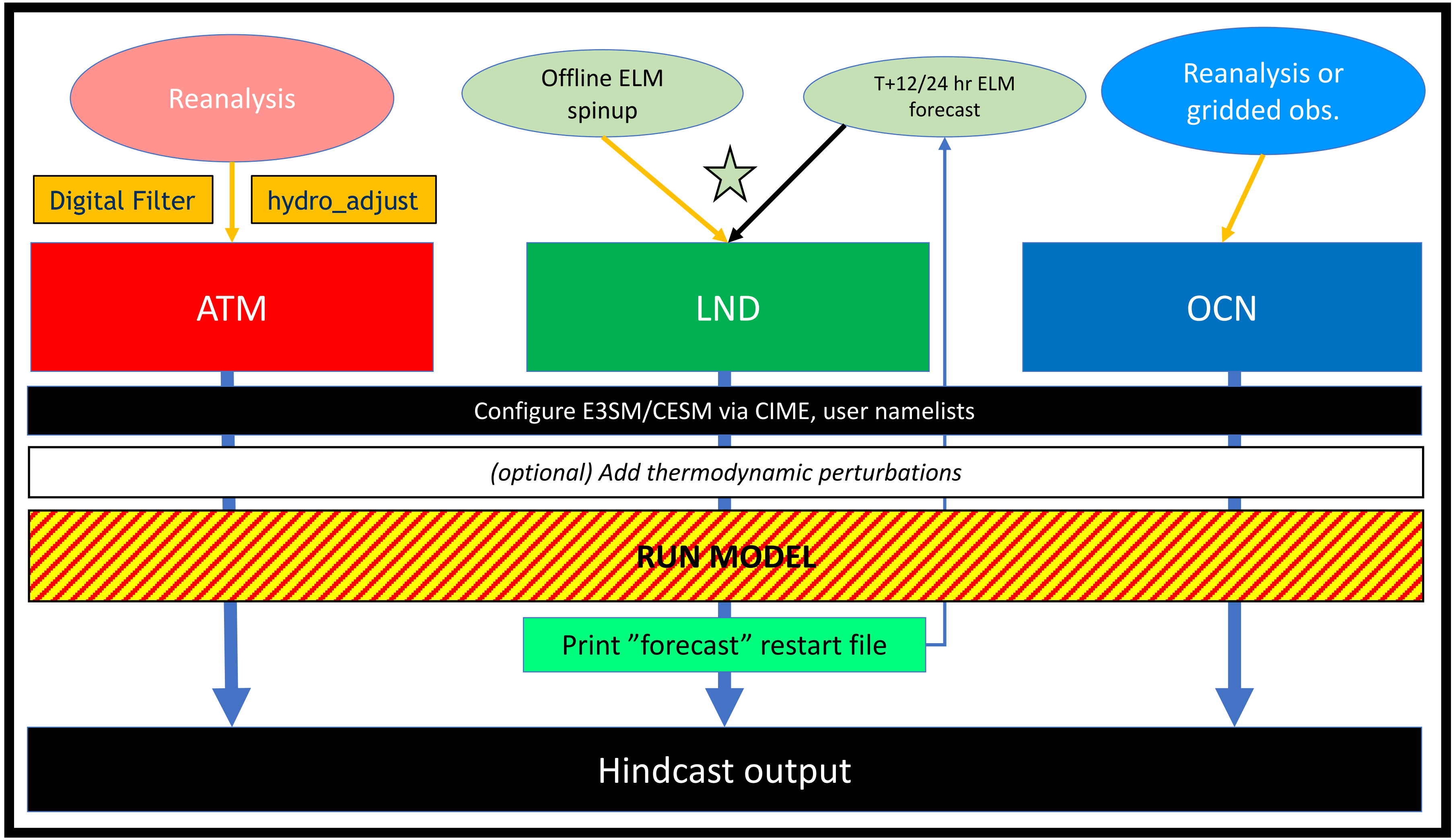


Qian et al., 2018

Betacast: hindcast support for CESM

PennState

https://github.com/zarzycki/betacast



Zarzycki and Jablonowski, 2015, MWR

(optional) TC tracking and statistics

The process?

PennState

Generate set (P) of parameter inputs and ranges

```
lubb_C_invrs_tau_bkgnd,0.1,5.0
clubb_C_invrs_tau_sfc,0.01,2.0
clubb_C_invrs_tau_shear,0.01,2.0
clubb_C_invrs_tau_N2,0.005,1.0
clubb_gamma_coef,0.1,0.9
clubb_gamma_coefb,0.1,0.9
clubb_c11,0.1,0.9
clubb_c11b,0.1,0.9
clubb_c8,2,8
clubb_beta,1.0,3.0
clubb_c_uu_shr,0,1
clubb_c_uu_buoy,0,1
clubb_c_invrs_tau_n2_clear_wp3,
clubb_c_invrs_tau_n2_wp2,0,1.2
clubb_c_invrs_tau_n2_xp2,0,0.7
clubb_c_invrs_tau_n2_wpxp,0,0.05
clubb_c_invrs_tau_wpxp_ri,0,1.0
clubb_altitude_threshold,25.0,1000.0
clubb_up2_sfc_coef,0,10
zmconv_tau,1800.,14400.
zmconv_ke,0.5e-6,10.0e-6
zmconv_c0_ocn,0.001,0.008
se_nu,0.20e15,1.0e15
```

P = 23

Generate LHS sample of size Z

```
# Do LHS calculations
sampler = qmc.LatinHypercube(d=num_vars)
#optimization="random-cd" added in 1.8.0
#sampler = qmc.LatinHypercube(d=num_vars, optimiz
sample = sampler.random(n=num_samples)
scaled_sample = qmc.scale(sample, l_bounds, u_bou
```

Z = 1200

Resulting set of combinations (P lines, N columns)

```
2.235948e+00,1.554339e-02,1.952923e-01,9.122890e-01,2.279078e-01,5.57633
+02,4.783025e+00,5.193065e+03,4.999672e-06,4.165975e-03,3.956101e+14
3.534685e+00,1.608991e+00,5.220681e-01,9.069053e-01,4.120342e-01,3.70315
+02,5.741669e+00,3.907741e+03,7.819348e-06,5.576695e-03,9.961181e+14
1.705845e+00,3.213369e-01,1.386629e+00,4.559371e-01,8.768336e-01,4.25828
+02,5.133134e+00,1.380007e+04,9.195500e-06,7.161500e-03,7.955536e+14
2.072631e-01,1.503143e+00,2.491256e-02,4.590301e-01,1.559282e-01,2.09377
+02,2.995484e-01,2.404167e+03,2.502762e-06,5.110206e-03,7.337338e+14
1.130345e+00,9.305662e-01,1.616388e+00,9.387361e-02,2.904719e-01,7.20795
+02,1.640975e+00,5.032015e+03,7.666990e-06,7.470773e-03,7.911111e+14
1.679859e+00,1.175868e+00,9.201679e-01,9.885945e-01,8.134322e-01,8.04374
+02,5.218515e+00,1.051964e+04,8.461415e-06,3.343271e-03,6.808574e+14
9.343254e-01,4.103287e-01,8.032468e-01,1.136831e-01,4.711610e-01,6.30015
+02,9.459529e+00,3.936652e+03,6.452451e-06,6.152243e-03,7.255034e+14
2.281685e+00,1.522456e+00,5.030232e-02,8.805720e-01,7.752457e-01,2.15028
+02,4.948609e+00,3.441481e+03,9.873246e-06,7.043738e-03,4.584735e+14
@
"param_values.csv" 1200L, 358800B
```

Each line is a single set of parameters to pass in via user_nl_cam

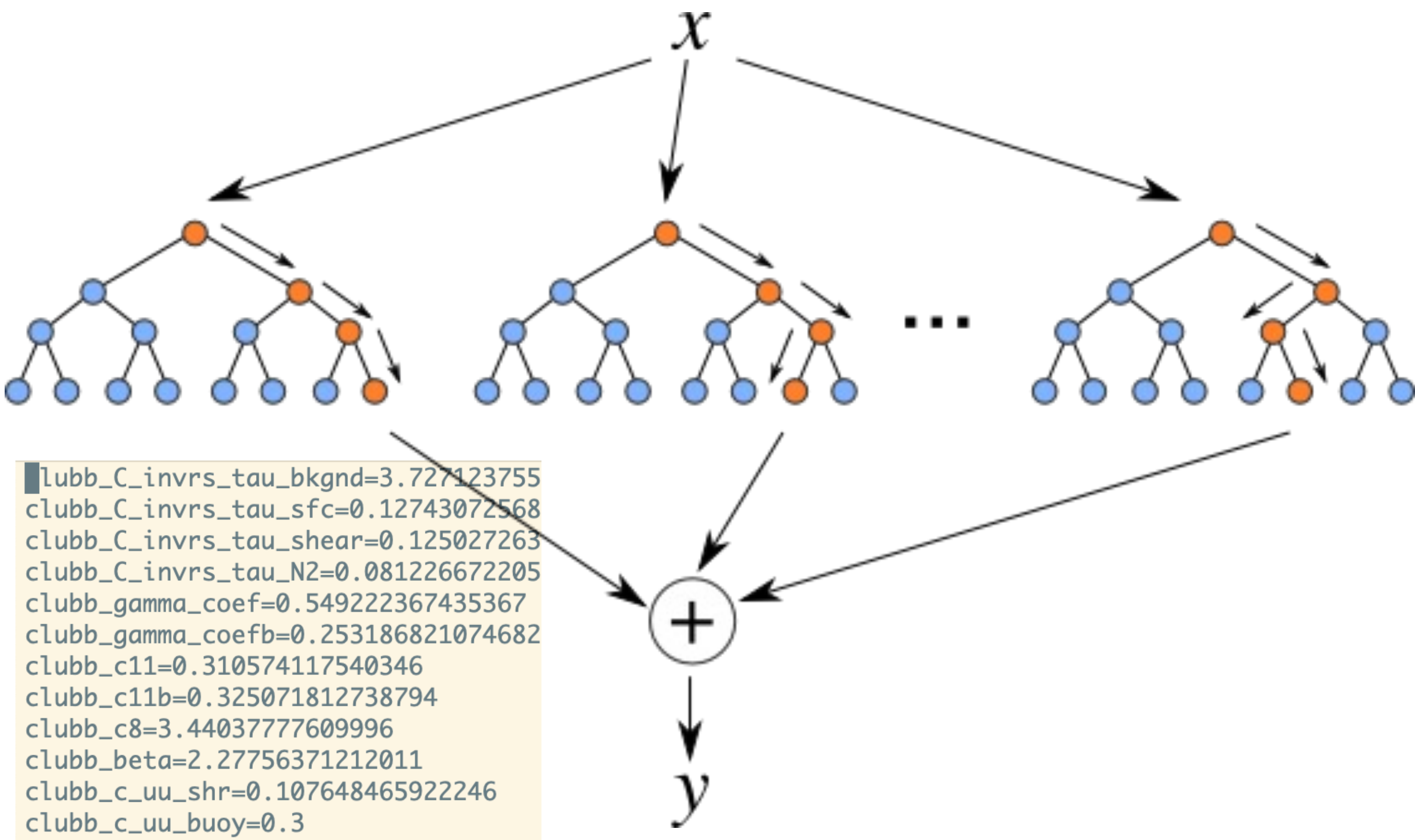
The mechanics?

PennState

- Pick N_inits (6, JMMJSN) that each run for D (3) days
- So we have Z * N_inits = 7,200 simulations to run
- Parallelize on Cheyenne, not so bad! ~3-4 "real" days if spread out across 20 jobs
- Average output fields +24-36 hours for climate metrics, compute anomaly relative to reference (i.e., benchmark)
  - Currently aggregating to 5x5 grid boxes, although more of a practicality
  - Results in 1200 "outputs" which are *nlat* x *nlon* array for each CAM variable (1200 x 2263 = 2.7M points)

The tool?

- Train output "vectors" (flattened 2-D arrays) using *random forest regressor*
- Model can be run with user editing a user_nl_cam (i.e., x vector)
- Model predicts *anomaly* vector relative to *reference*
- Reshape back to 2-D array for analysis!

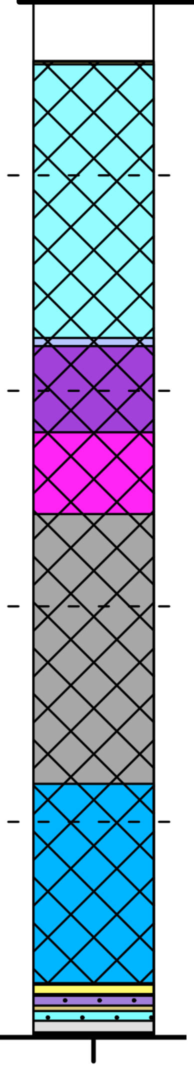



```
lubb_C_invrs_tau_bkgnd=3.727123755
clubb_C_invrs_tau_sfc=0.12743072568
clubb_C_invrs_tau_shear=0.125027263
clubb_C_invrs_tau_N2=0.081226672205
clubb_gamma_coef=0.549222367435367
clubb_gamma_coefb=0.253186821074682
clubb_c11=0.310574117540346
clubb_c11b=0.325071812738794
clubb_c8=3.44037777609996
clubb_beta=2.27756371212011
clubb_c_uu_shr=0.107648465922246
clubb_c_uu_buoy=0.3
clubb_c_invrs_tau_n2_clear_wp3=1.0
clubb_c_invrs_tau_n2_wp2=0.1
clubb_c_invrs_tau_n2_xp2=0.8
clubb_c_invrs_tau_n2_wpxp=0
clubb_c_invrs_tau_wpxp_ri=0.35
clubb_altitude_threshold=100
clubb_up2_sfc_coef=2.0
zmconv_tau=3600
zmconv_ke=5.0E-6
zmconv_c0_ocn=0.0300
```

Interpretable: what is important in making a prediction?

PennState

Qian et al., 2018

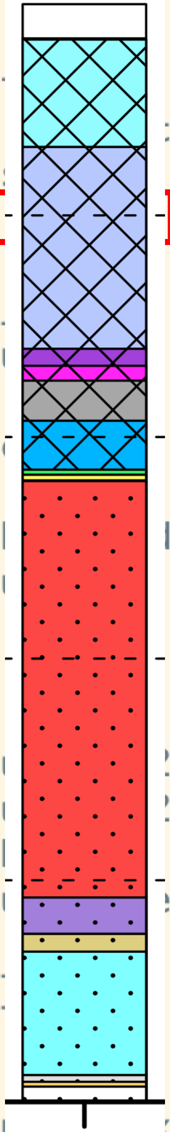


CLDLOW

```
{'bootstrap': Tr                 ': 0.0, 'criterion': 'squ
split': 2, 'min_weight_fraction_leaf': 0.0, 'n_estimator
0.21145597135218605     ['clubb_C_invrs_tau_shear']
0.12463443474785157     ['clubb_c8']
0.12342257234142996     ['clubb_c_uu_shr']
0.07391954146511641     ['clubb_gamma_coef']
0.06919068386443027     ['clubb_c_invrs_tau_n2_xp2']
0.0577467637024121     ['clubb_c_invrs_tau_n2_wp2']
0.05464471274454084     ['clubb_up2_sfc_coef']
0.049369764776027704     ['clubb_c11b']
0.030048289208605573     ['clubb_c11']
0.02360788463659788     ['clubb_C_invrs_tau_N2']
0.022327059578247193     ['clubb_C_invrs_tau_sfc']
0.018762494092775136     ['clubb_gamma_coefb']
0.018036163351335226     ['clubb_C_invrs_tau_bkgnd']
0.016158214378506144     ['clubb_c_invrs_tau_wpxp_ri']
0.015510901361020353     ['zmconv_tau']
0.014994598952002312     ['zmconv_c0_ocn']
0.012494952361274148     ['clubb_altitude_threshold']
0.01215
0.01075
0.0103
0.0103
0.0101
0.009995213743247003     ['clubb_c_uu_buoy']
MAE is 0.023259240956448914
```

PRECT

```
RF
{'bootstrap': True             a': 0.0, 'cri
split': 2, 'min_weight_fraction_leaf': 0.0
0.3660887501511995     ['clubb_C_invrs_tau_
0.2320714694982504     ['zmconv_tau']
0.1027427176794276     ['clubb_c_uu_shr']
0.03815391055362532     ['clubb_C_invrs_tau
0.031082624577777974     ['clubb_C_invrs_tau
0.019294462641779916     ['zmconv_ke']
0.018479361177184808     ['clubb_up2_sfc_co
0.017884228805207063     ['clubb_c8']
0.014472660114623509     ['clubb_altitude_t
0.013667826436264269     ['clubb_C_invrs_tau
0.013630740292987121     ['clubb_gamma_coef
0.012188004150956996     ['se_nu']
0.0116272685909559     ['clubb_c11b']
0.011555666832940022     ['clubb_c_invrs_tau
0.011453657677168197     ['clubb_c_invrs_tau
0.011386226102659524     ['clubb_gamma_coef
0.011260959725820783     ['clubb_c_invrs_tau
0.009707215903686292     ['clubb_c_invrs_tau
MAE is 0.2078194178048861
```



- Benefit of RF, training provides ranking of sensitivity
- See Kyle Nardi's talk later today

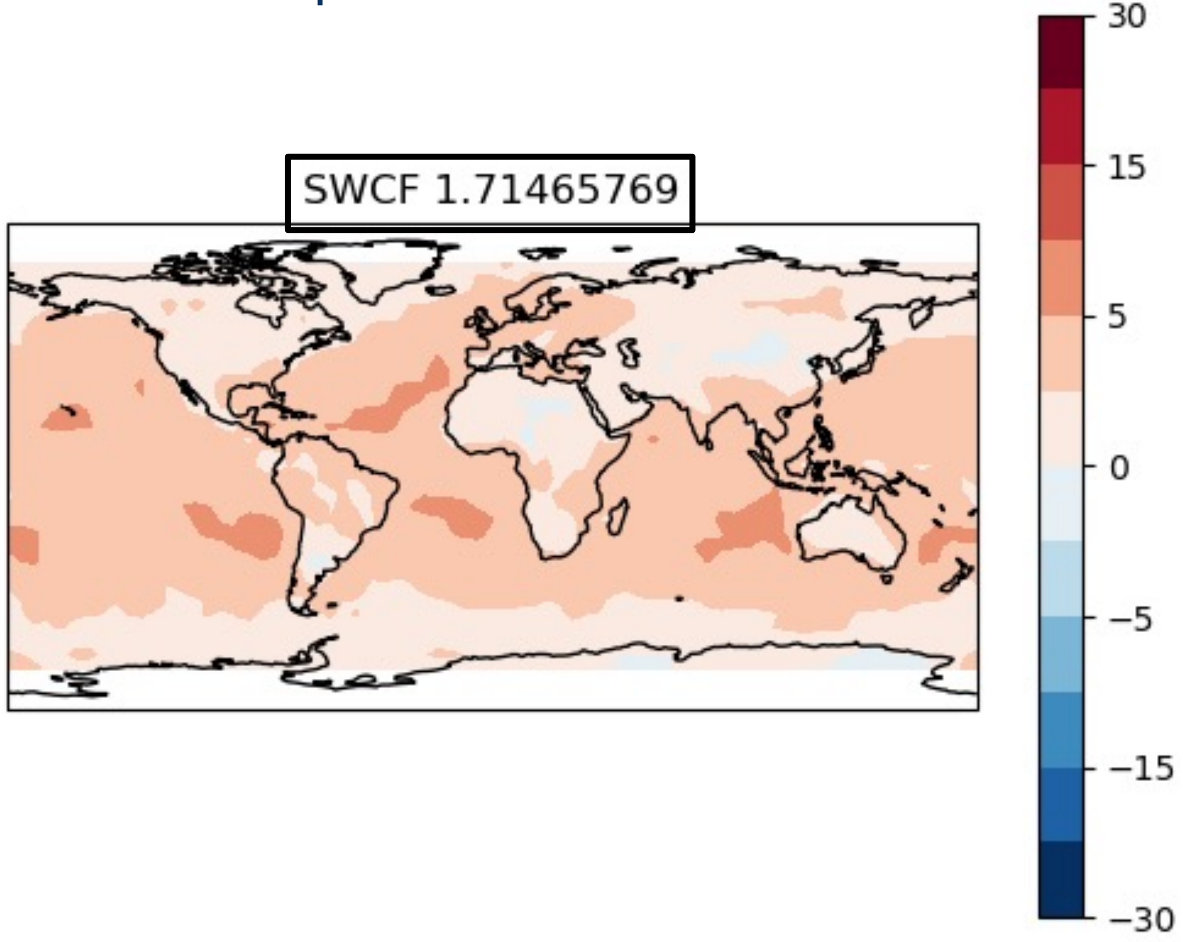
Reference -> perturbation

- Let's try changing a few things…
  - clubb_C_invrs_tau_shear=0.1 (from 0.12)
  - clubb_gamma_coef=0.5 (from 0.55)
  - clubb_c8=2.8 (from 3.4)
  - clubb_c_uu_shr=0.5 (from 0.1)
  - clubb_c_invrs_tau_n2_wp2=0.05 (from 0.1)
  - clubb_c_invrs_tau_n2_xp2=2.0 (from 0.8)
  - clubb_altitude_threshold=20 (from 100)

SWCF prediction

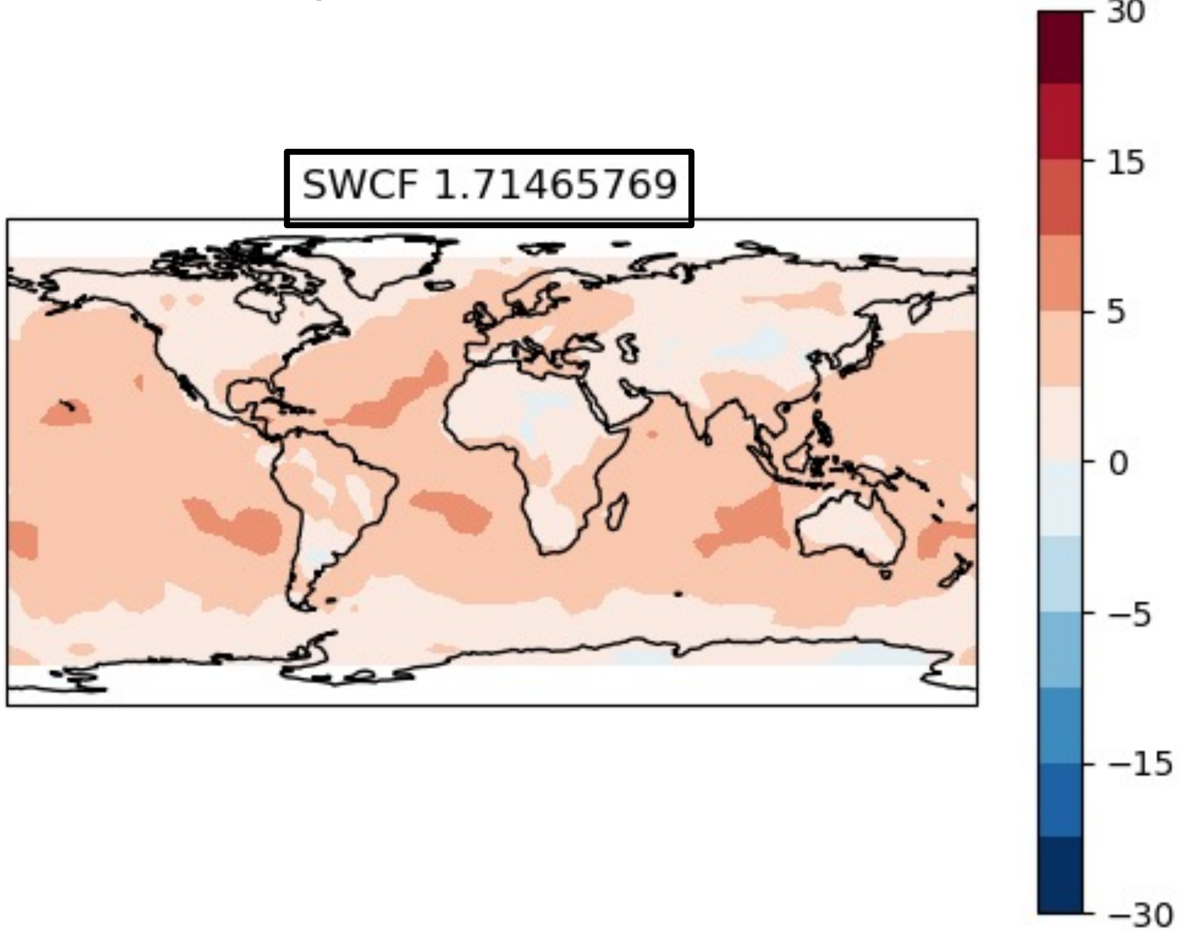
PennState

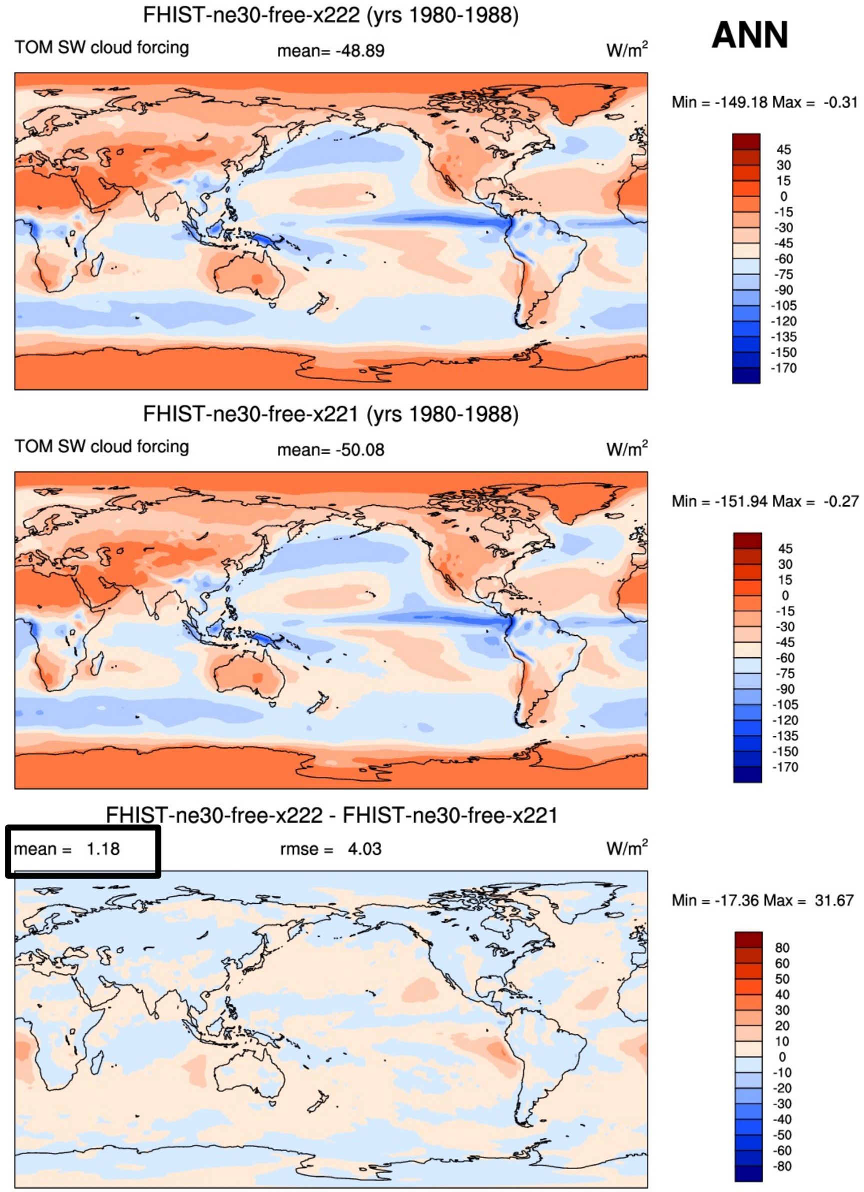
- Emulator predicts…



SWCF prediction

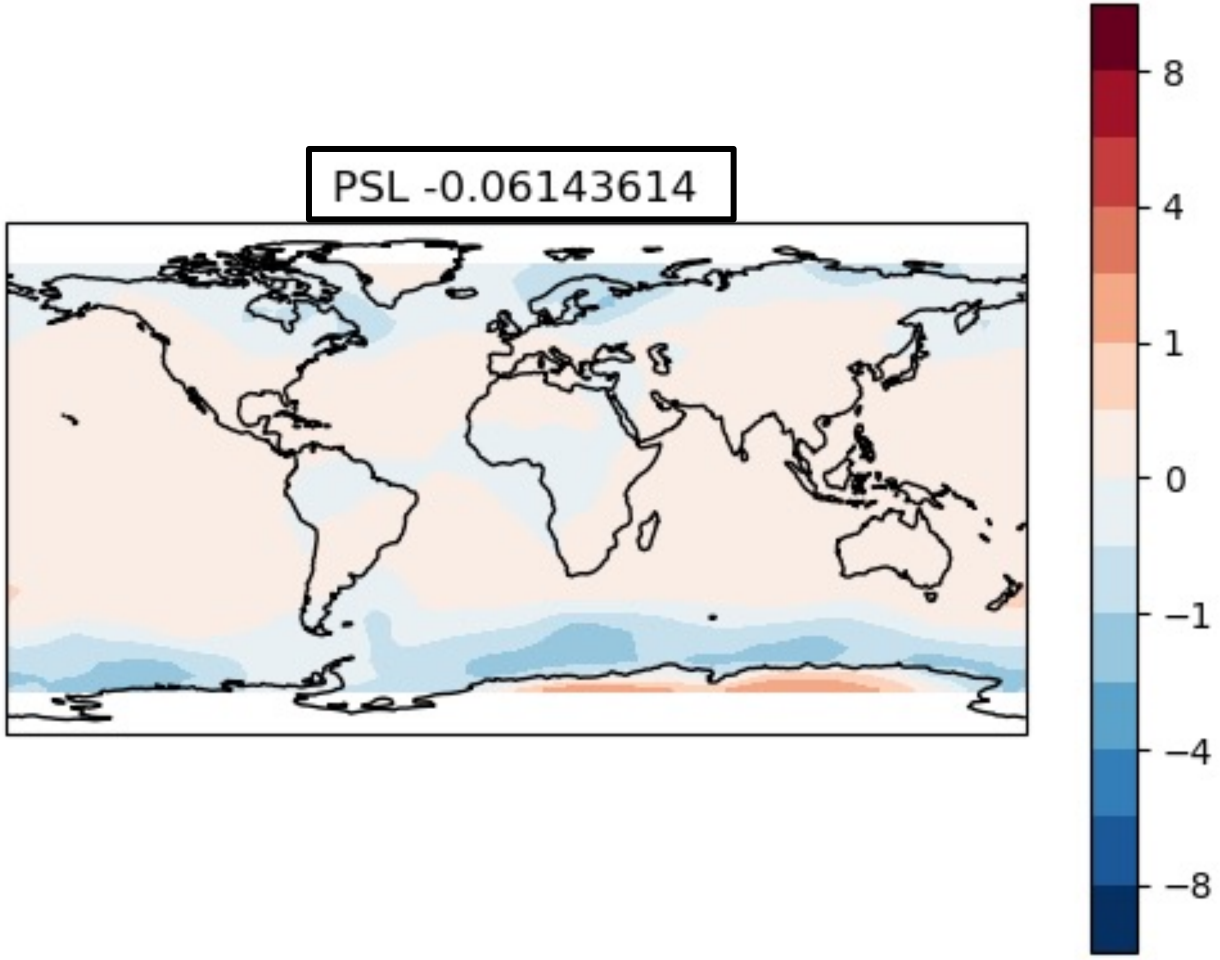
- Emulator predicts…

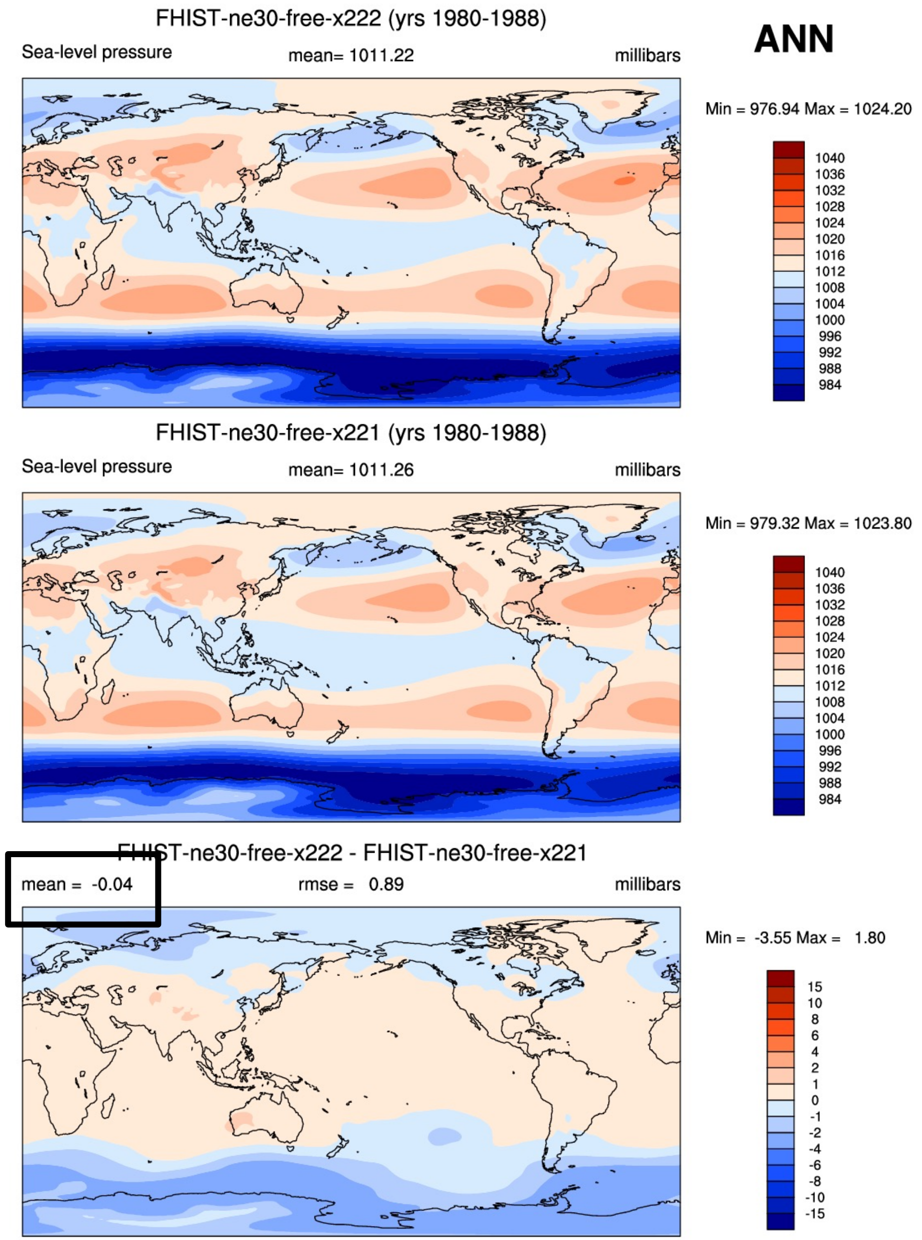




Note, color bars don't match!

PSL prediction

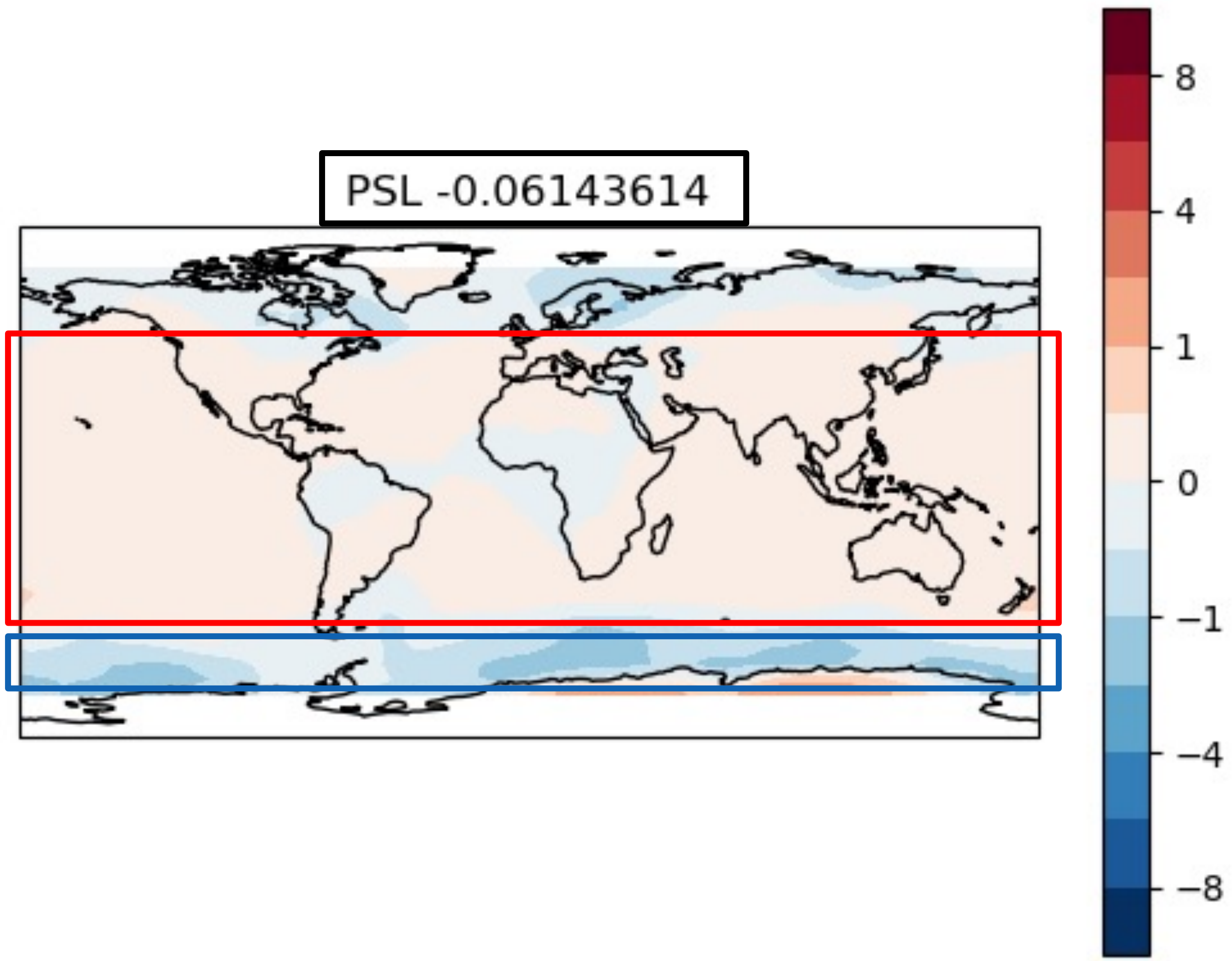


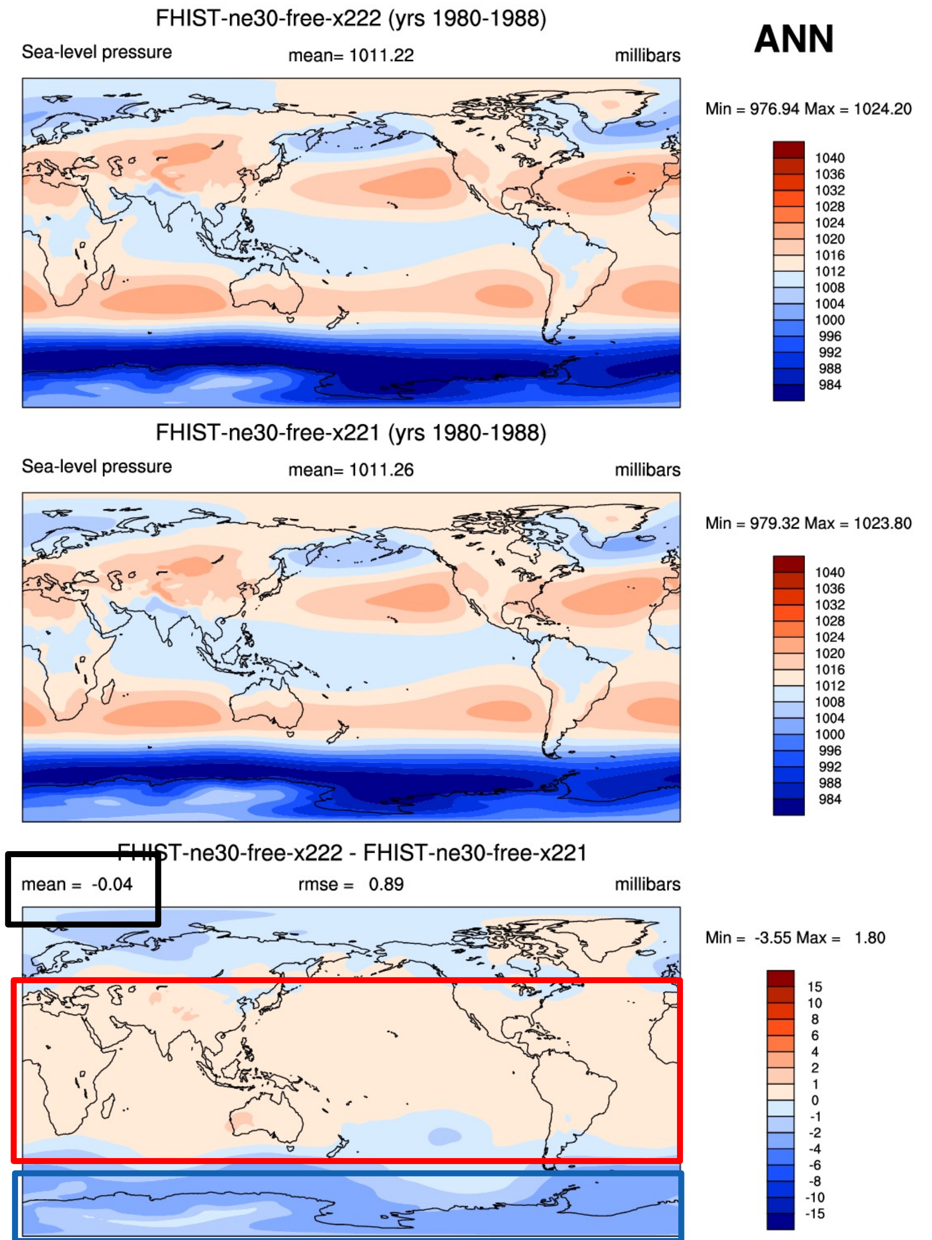


Note, color bars don't match!

PSL prediction

PennState





Note, color bars don't match!

Cost?

- Cost: we have Z * N_inits * D days we have to integrate CAM for
  - Here, 21,600 days =~60 year single member
  - Runs are small, help fill "economy" backfill!
- Outputs distilled into climatology files from CAM output O(1GB)
- Training for each var ~15 seconds on Macbook Pro (longer for XGBoost and NNs, but not otherworldly) → can be "pickled"
- Each "model" var takes ~8 seconds to run

Summary

PennState

- First blush?
  - Excellent for global mean *directional signal*!
  - Not bad for global mean *magnitude* (?)
  - Hint of regional information, but unclear (???)
  - Fast!
- Next steps:
  - Code cleanup / organization
  - Automatic optimization
  - Incorporate information about CLUBB booleans
  - New deck with expanded LHS using 6_3_091
- AMWG questions?
  - How to better leverage short runs for development + tuning?
  - Can this provide alternative pathways for physical interpretability?