Correcting Storm Displacement Errors in Ensembles Using the Feature Alignment Technique (FAT)

DEREK STRATMAN, OU/CIMMS AND NOAA/OAR/NSSL

COREY POTVIN, OU/CIMMS AND NOAA/OAR/NSSL

LOUIS WICKER, NOAA/OAR/NSSL

Motivation

A goal of *Warn-on-Forecast* (WoF) is to develop forecasting systems that produce accurate analyses and forecasts of severe weather for operational warning settings.

Recent WoF-related studies have noticed or indicated the need to alleviate *storm displacement errors* in both storm-scale analyses and short-term forecasts.

Storm displacement errors can develop from various sources.

Promising solution: Feature Alignment Technique (FAT)

What is the FAT?

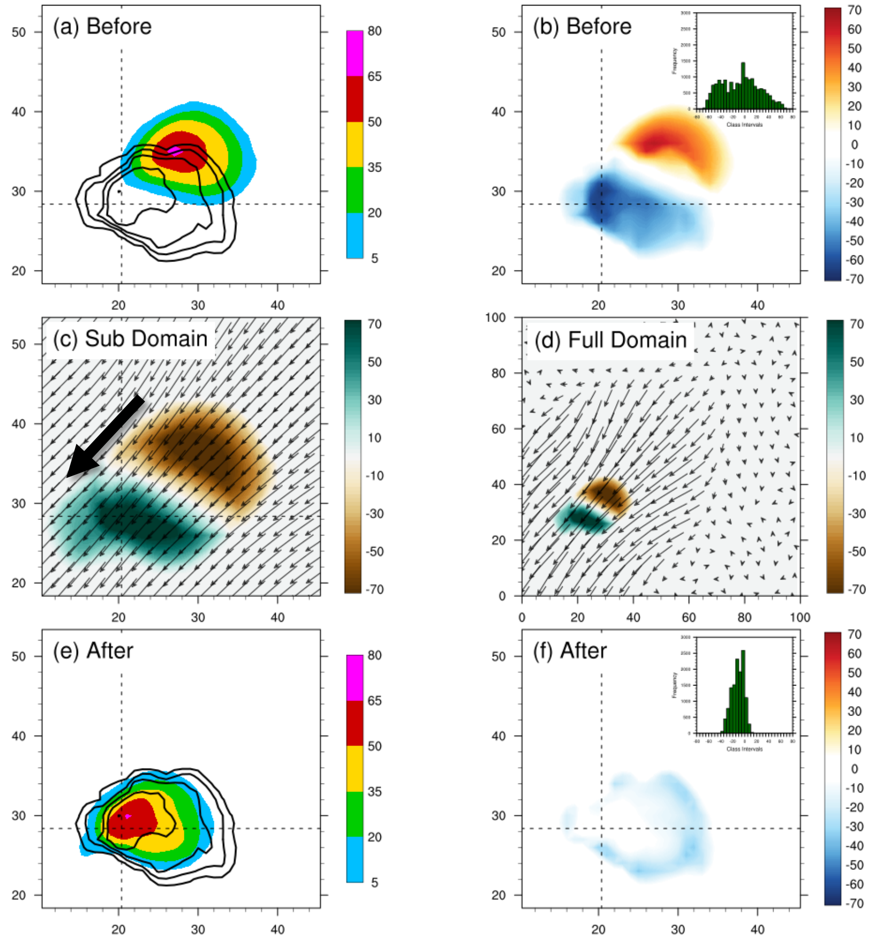
FAT (Nehrkorn et al. 2015) minimizes a cost function for observed and forecasted fields to determine a 2-D field of displacement vectors.

- e.g., radar reflectivity and/or IR satellite

Cost function is composed of a residual error function (J_r) and penalty terms, including:

- Smoothness (J_s)
- Divergence (J_d)
- Magnitude (J_m)
- Barrier (J_b)

The FAT is performed prior to data assimilation steps.



OSSEs using CM1

OSSEs performed on 2-km grid.

- 100x100x50 gridpoint domain
- 50-member ensembles

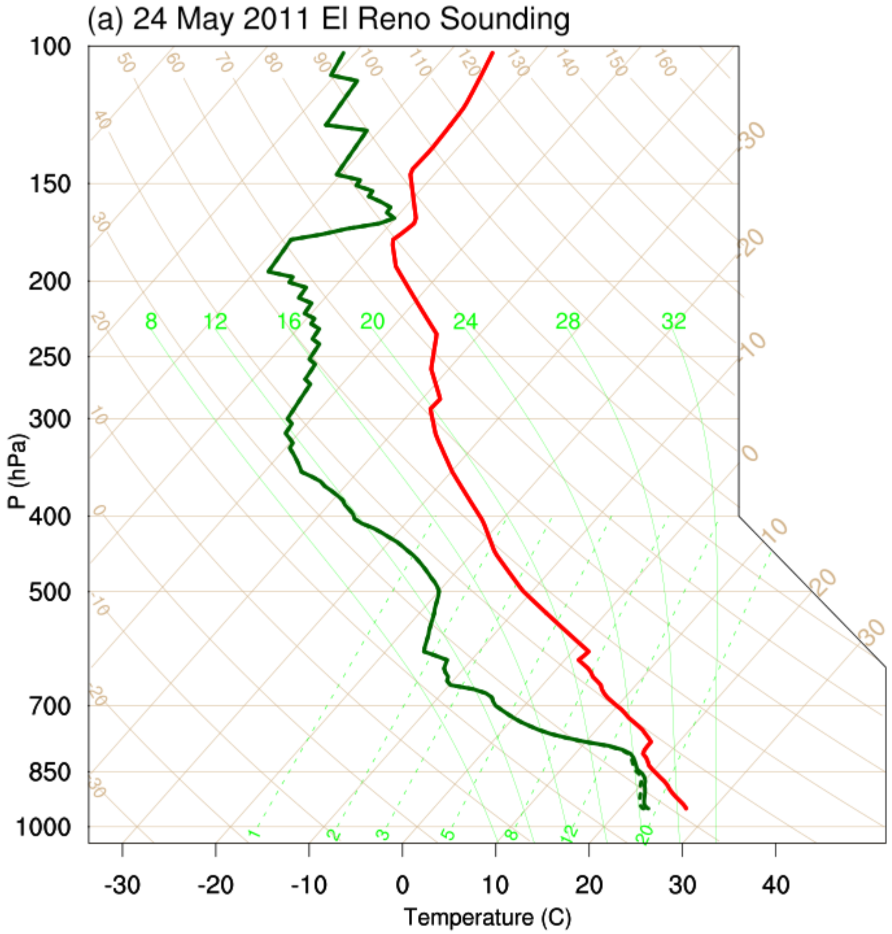
Pseudo-radar observations provided by 250-m **Truth** run of a supercell.

- 800x800x50 gridpoint domain
- *Thompson* microphysics

Initialized from thermal bubbles.

Environment provided by ”El Reno” sounding.

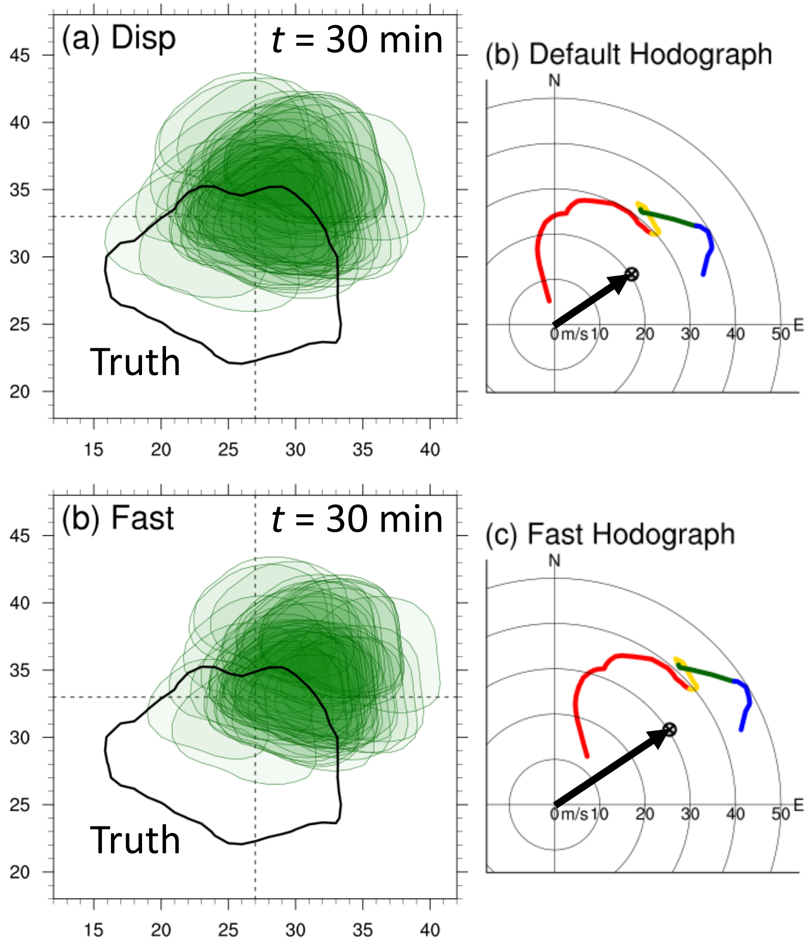
Local Ensemble Transform Kalman Filter (LETKF) for data assimilation.



Perfect and Imperfect OSSEs

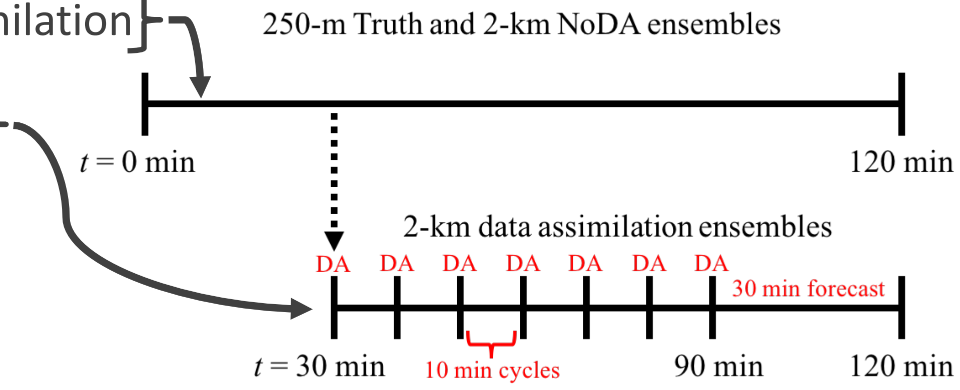
Four experiments:

- **DispTh** – *initial displacement* ~14 km to the NE and *Thompson*
- **DispLG** – *initial displacement* ~14 km to the NE and *LFO-Goddard*
- **FastTh** – sounding with 10 m/s *faster storm motions* and *Thompson*
- **DispAdjqvLG** – same as **DispLG** except *decreased moisture* < 2-km AGL

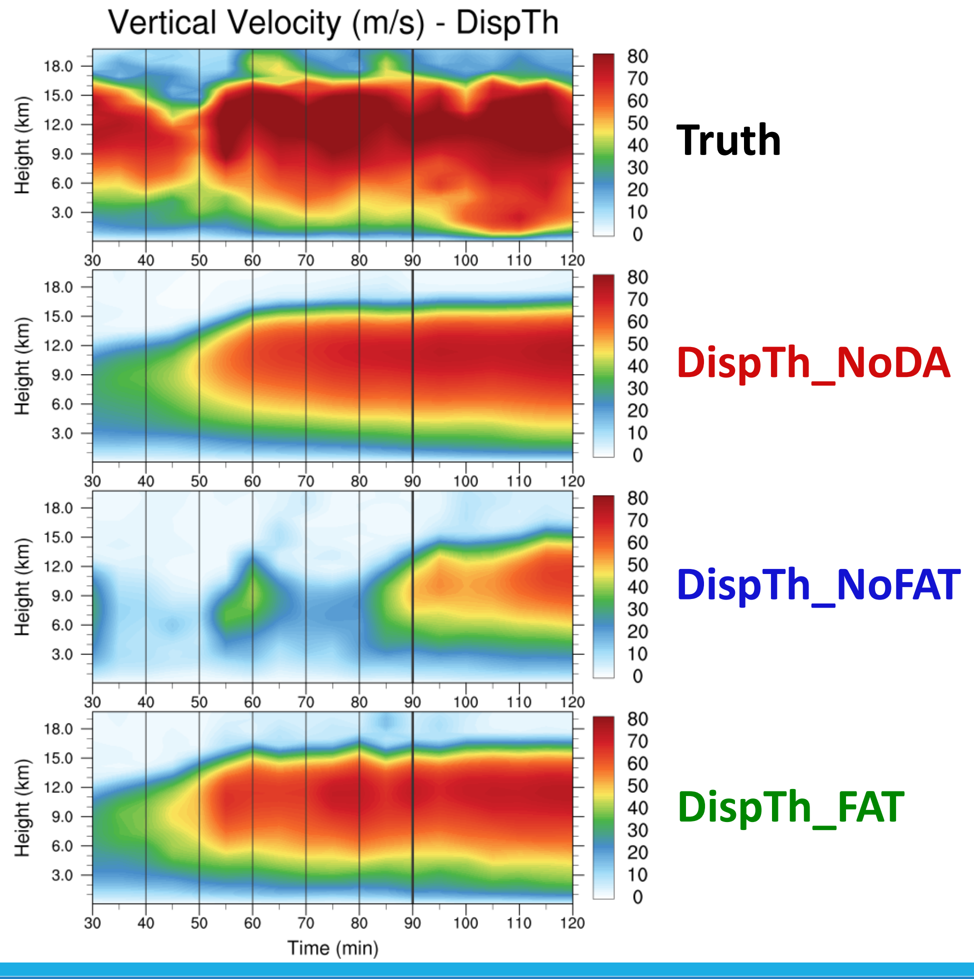


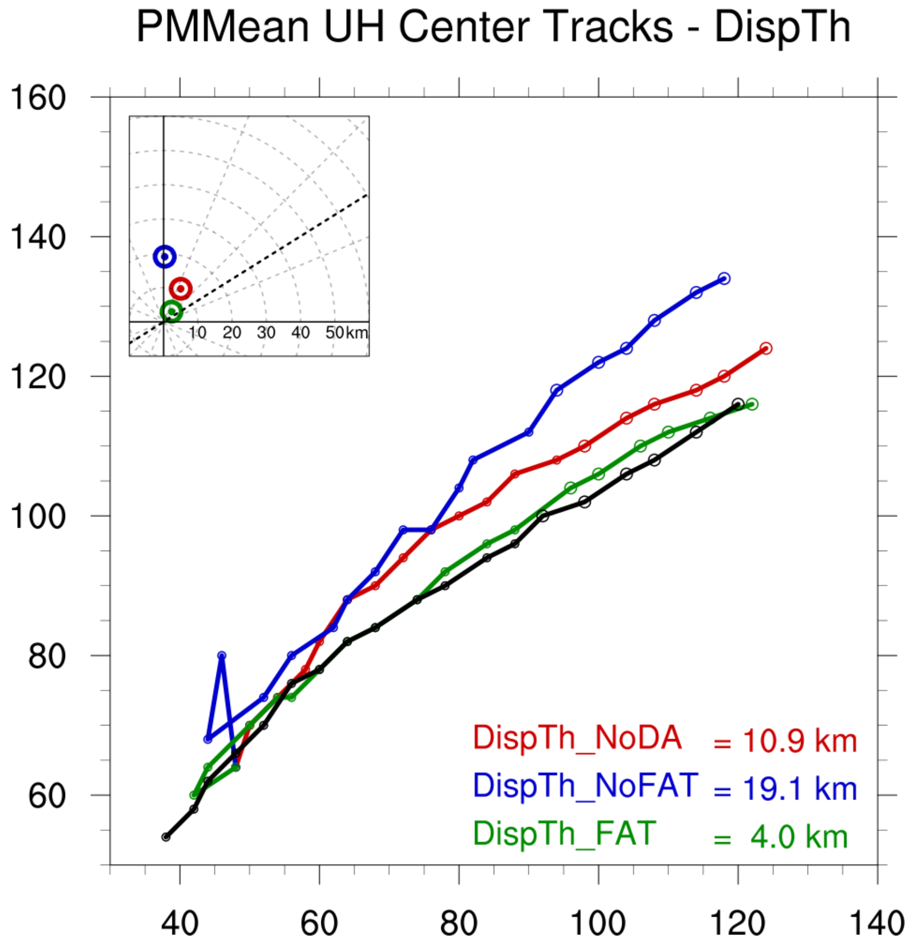
Three ensembles for each OSSE:

- **NoDA** – no data assimilation
- **NoFAT** – only LETKF
- **FAT** – FAT and LETKF

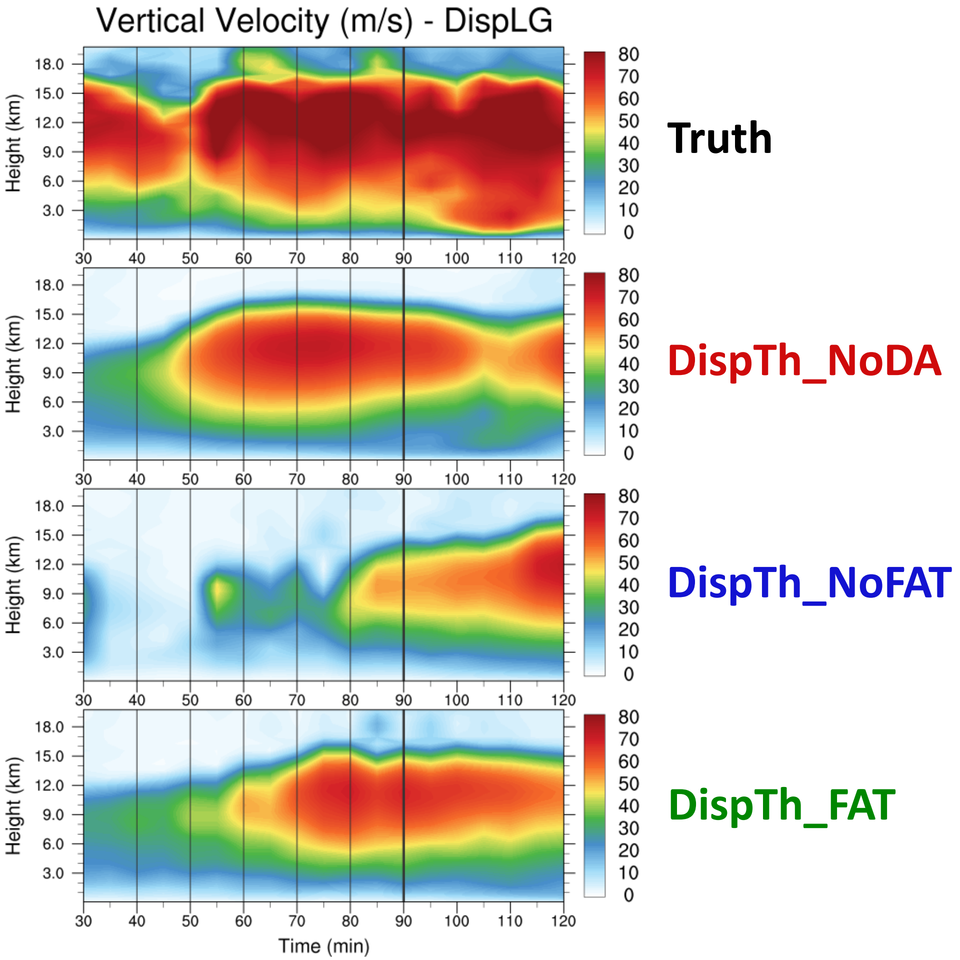


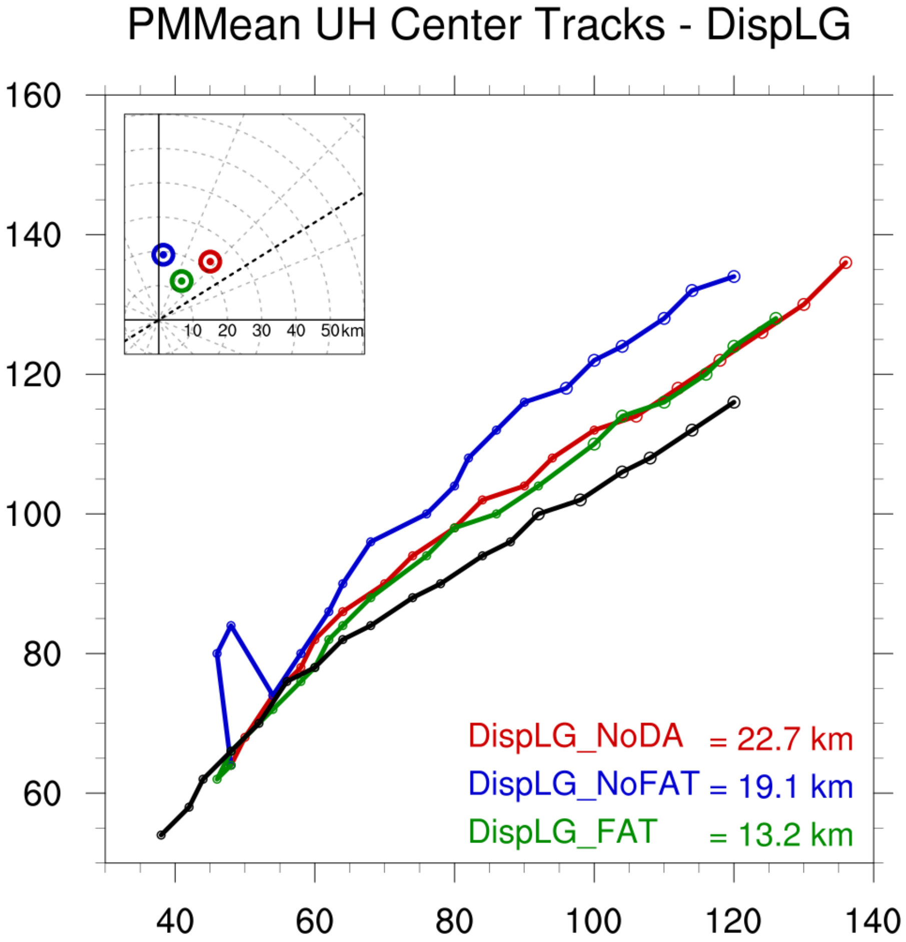
Initial Displacement and Thompson



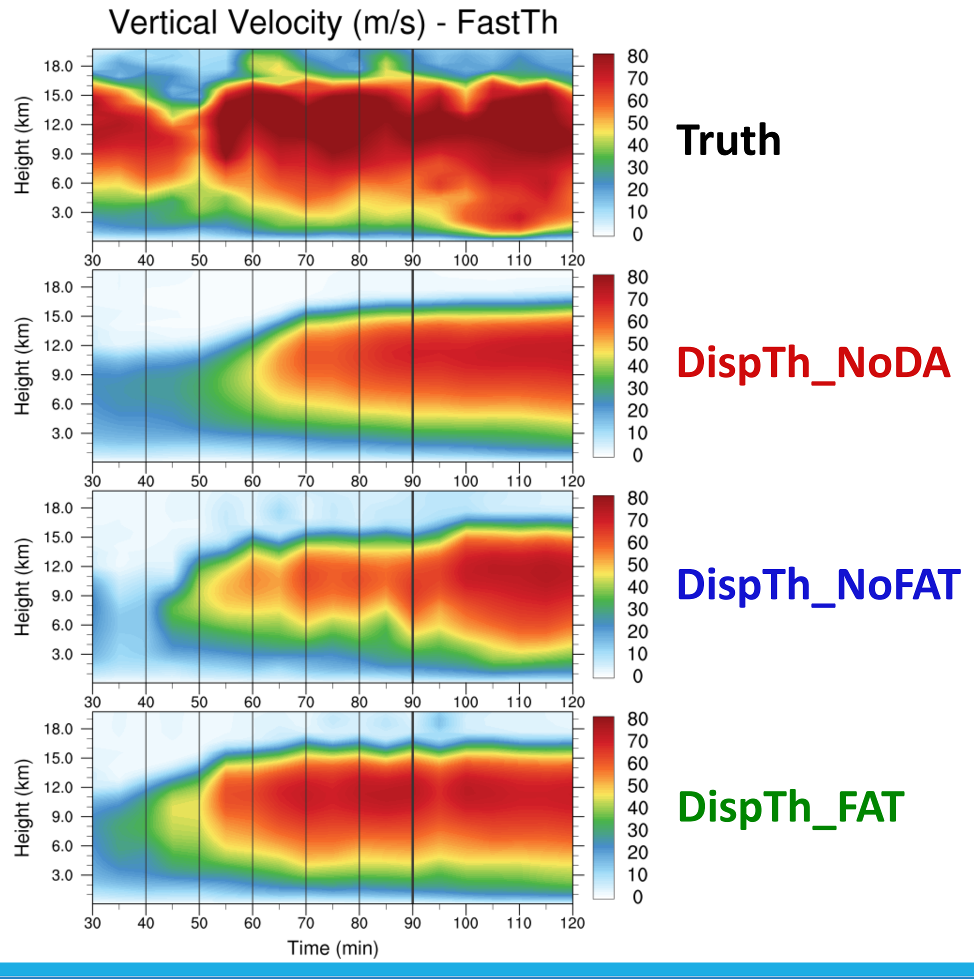


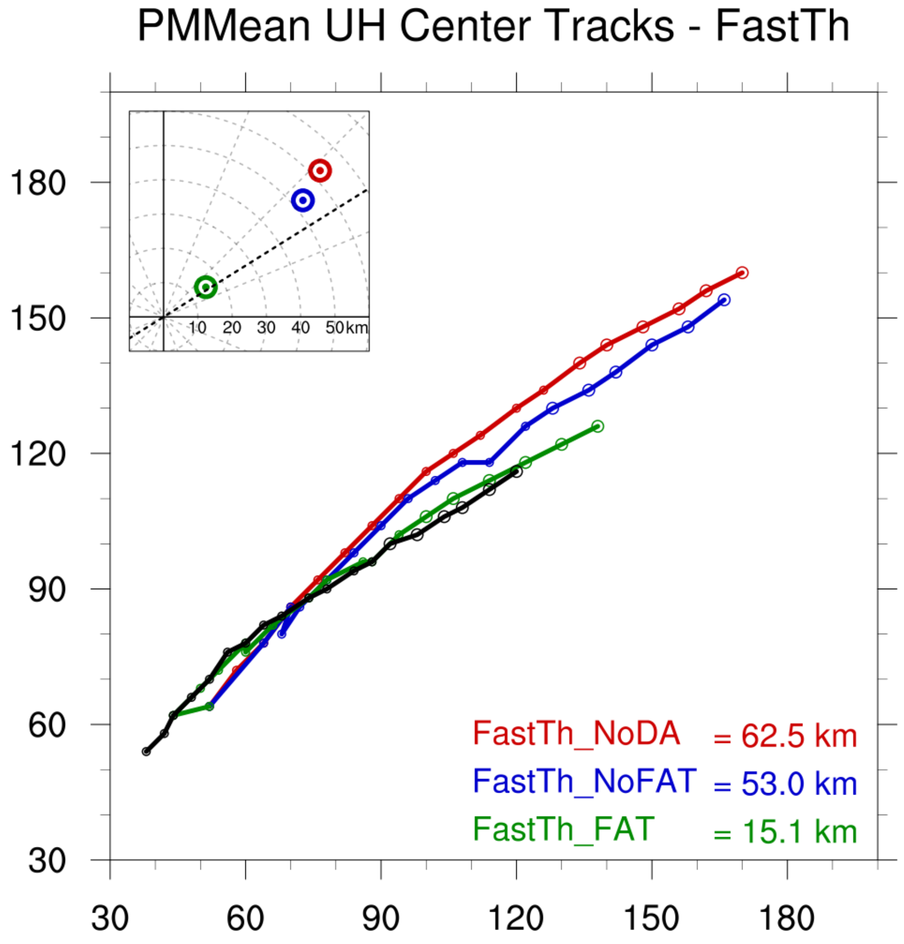
Initial Displacement and LFO-Goddard



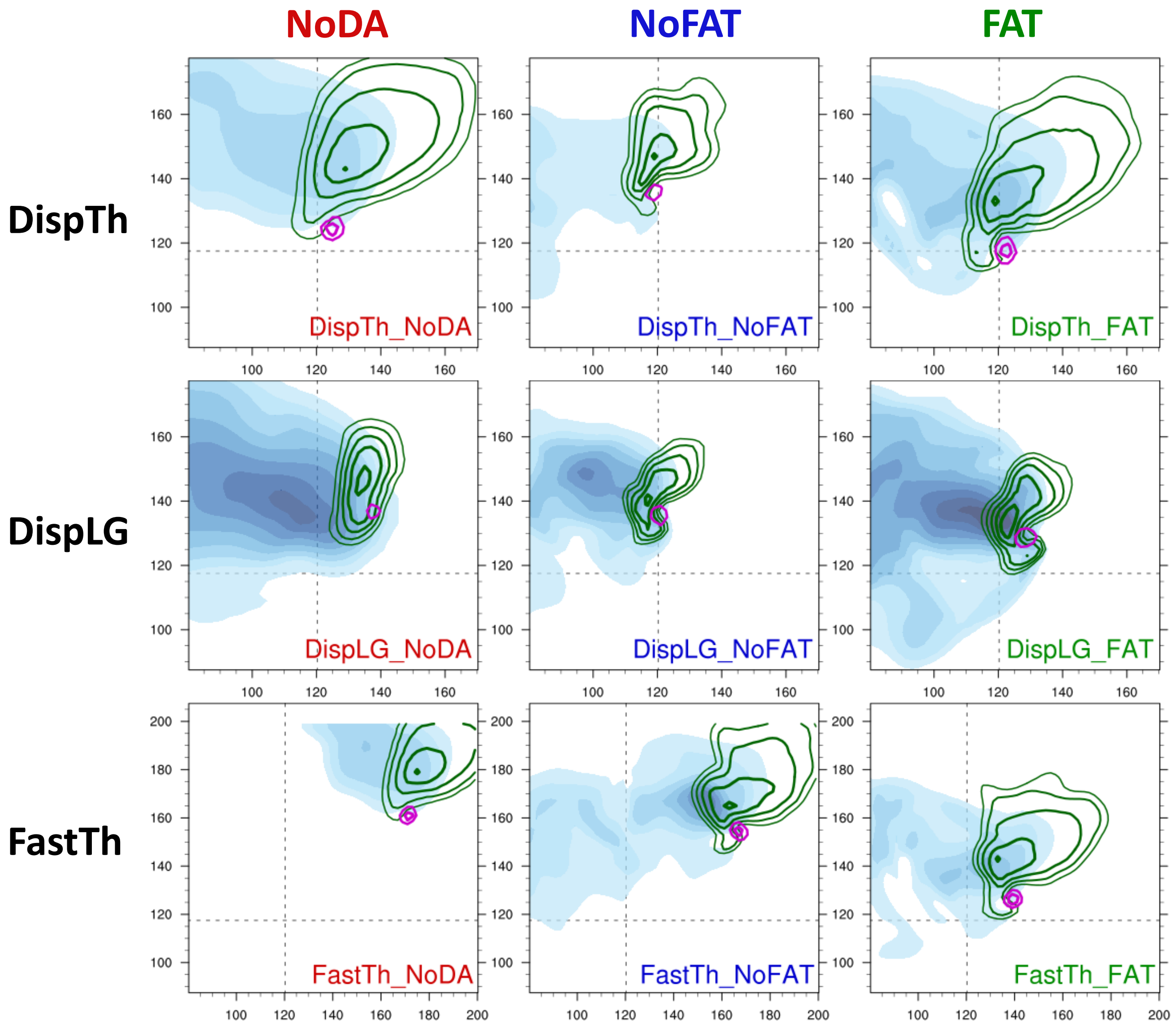


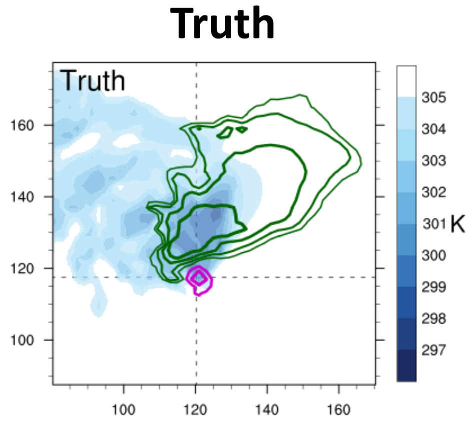
Faster Storm Motion and Thompson





30-min Forecasts





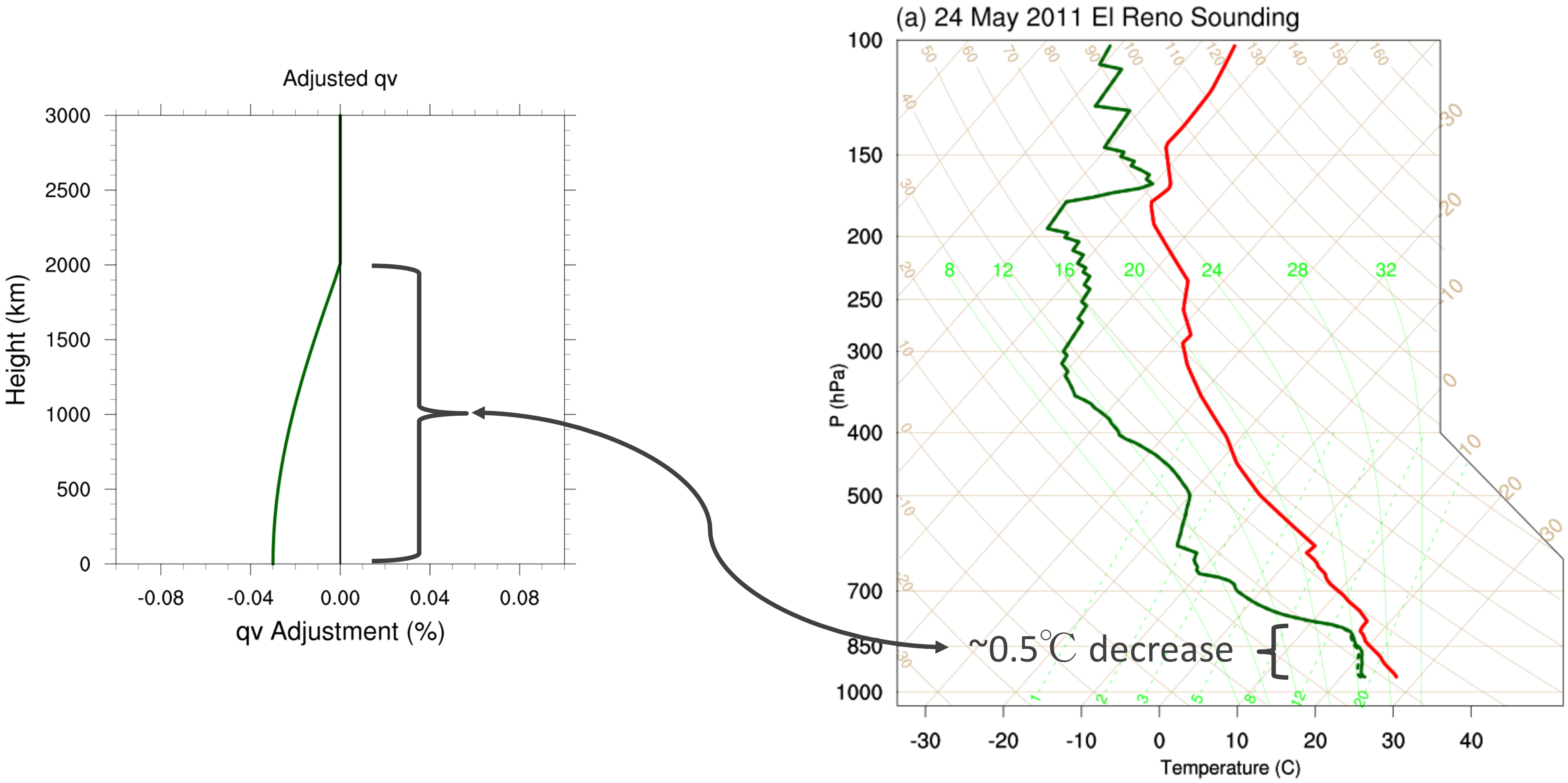
Probability Matched Mean fields:

Near-surface potential temperature from 297–305 K

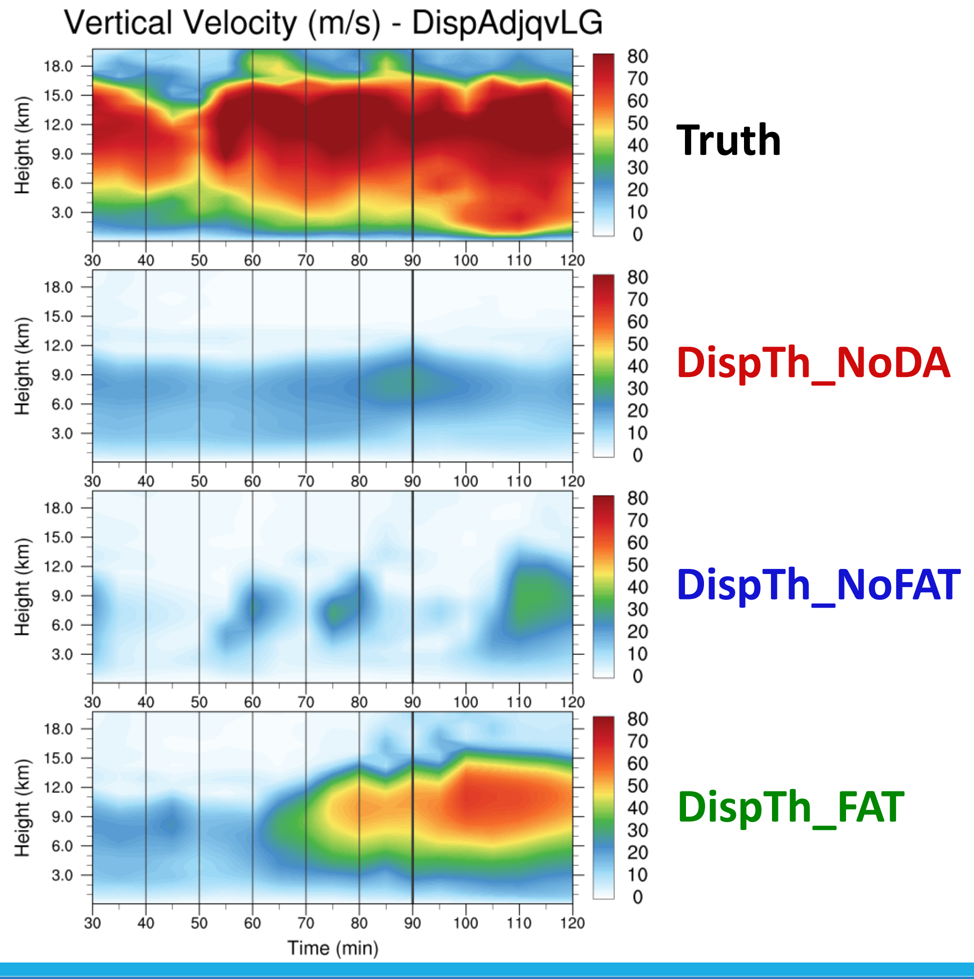
1-km reflectivity at 20, 30, 40, 50, and 60 dBZ

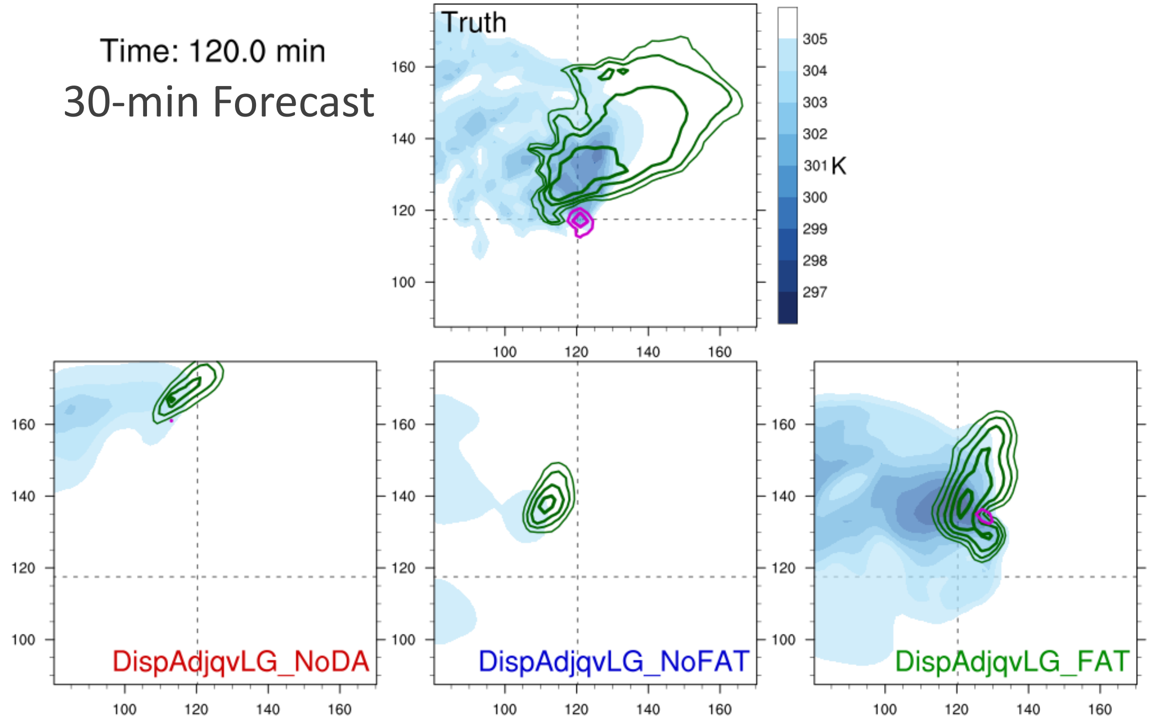
2–5-km updraft helicity at 500 and 2000 $m^2\ s^{-2}$

Drier Boundary Layer and LFO-Goddard



Drier Boundary Layer and LFO-Goddard





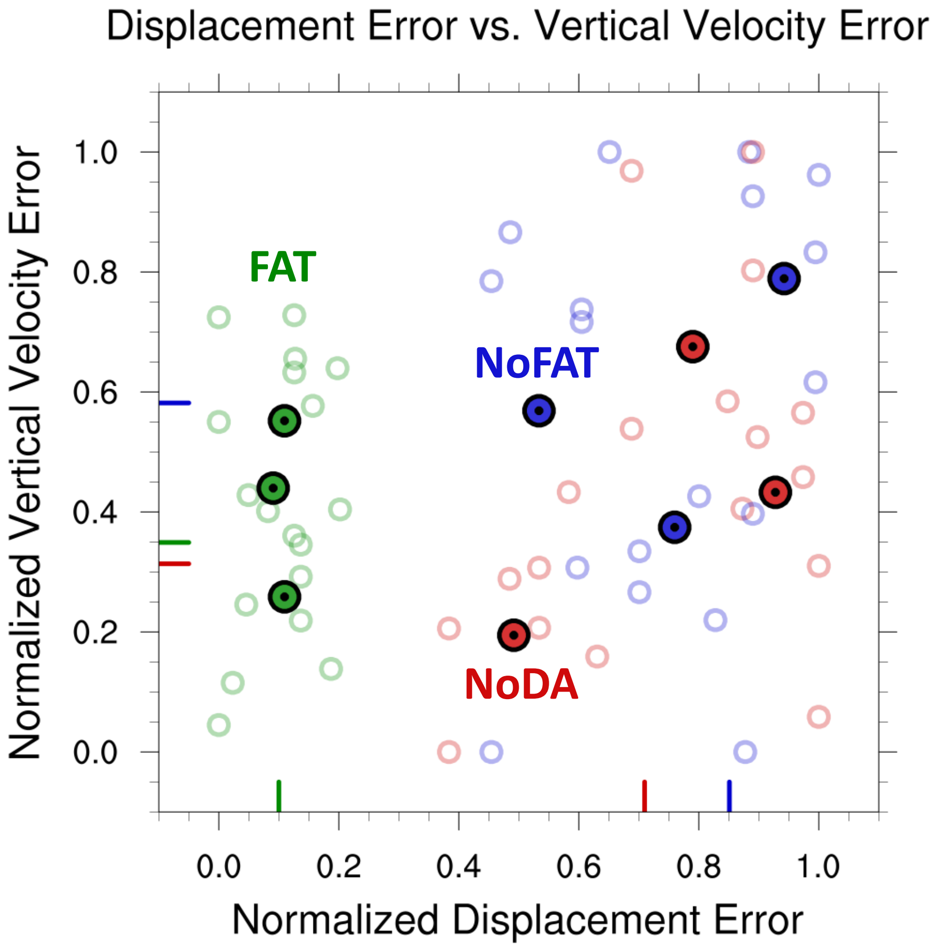
Probability Matched Mean fields:
Near-surface potential temperature from 297–305 K
1-km reflectivity at 20, 30, 40, 50, and 60 dBZ
2–5-km updraft helicity at 500 and 2000 $m^2\ s^{-2}$

Main Takeaways

These OSSEs have demonstrated that the **FAT** can potentially *improve* storm-scale ensemble analyses and short-term forecasts by substantially reducing storm displacement errors.

The **FAT** *improved* storm **intensities**, **spin-up times**, **locations**, and **structures** without introducing severe imbalances.

Also, the **FAT** decreased the sensitivity to storm-environment errors.



Next Steps for the FAT

OSSEs for multiple supercells, mixed-mode, and mesoscale convective system cases.

Incorporate terrain effects (as in Nehrkorn et al. 2014).

Real-data isolated supercell cases using full-model physics in a framework similar to NEWS-e (NSSL Experimental WoF System for ensembles).

Real-data mixed-mode cases using full-model physics.

Thank you!

Derek.Stratman@noaa.gov

FAT: Under the Hood

Nonlinear conjugate gradient algorithm

- scipy.optimize.fmin_cg

Smoothed composite reflectivity fields

- Gaussian filter with sigma = 15.0
- Thresholds of 0.9*max and 0.5*max
- Forecast CREF normalized to Obs CREF

Center of mass provides initial guess for CREF > 5 dBZ, otherwise random guesses

Vectors = 0 along domain edges

Vectors can't extend beyond edge of domain

Plethora of parameters:

- λ_s = 1.0; λ_d = 0.5; λ_m = 0.1; λ_b = 1.0
- S = 50 gu
- σ_o = 7.5 dBZ
- Δd = 1.2 gu and ΔL = 1.0 gu

$$\boldsymbol{a} = \delta\boldsymbol{i};\ \boldsymbol{b} = \delta\boldsymbol{j} \quad J = J_r[\boldsymbol{y}, H(\boldsymbol{x}), \boldsymbol{a}, \boldsymbol{b}] + J_p(\boldsymbol{a}, \boldsymbol{b})$$

$$J_r = \sum_{n=1}^{N_o} \frac{[y_n - x_n(i_n + a_n, j_n + b_n)]^2}{\sigma_{o,n}^2}$$

$$J_p = \lambda_s J_s + \lambda_d J_d + \lambda_m J_m + \lambda_b J_b$$

$$J_s = \frac{1}{\left(\frac{\Delta d}{(\Delta L)^2}\right)^2} \sum_{n=1}^{N_o} \left(\frac{\partial^2 a_n}{\partial i_n^2} + \frac{\partial^2 a_n}{\partial j_n^2}\right)^2 + \frac{1}{\left(\frac{\Delta d}{(\Delta L)^2}\right)^2} \sum_{n=1}^{N_o} \left(\frac{\partial^2 b_n}{\partial i_n^2} + \frac{\partial^2 b_n}{\partial j_n^2}\right)^2$$

$$J_d = \sum_{n=1}^{N_o} \left(\frac{\partial a_n}{\partial i_n} + \frac{\partial b_n}{\partial j_n}\right)^2$$

$$J_m = \sum_{n=1}^{N_o} \left(\frac{a_n}{S}\right)^2 + \sum_{n=1}^{N_o} \left(\frac{b_n}{S}\right)^2$$

$$J_b = \sum_{n=1}^{N_o} \left(\frac{a_n}{S}\right)^{20} + \sum_{n=1}^{N_o} \left(\frac{b_n}{S}\right)^{20}$$