Convective-Scale Ensemble Prediction Using Adaptive Gaussian/Non-Gaussian Ensemble Filters

Kenta Kurosawa
Jonathan Poterjoy

University of Maryland

9th NOAA Ensemble Users Workshop
Wednesday, 23rd August 2023

Motivation & Objectives

- **EnKFs:**

  Gaussian-based; biased in non-linear or non-Gaussian situations.

- **PFs:**

  Flexible with error distributions, but more sensitive to sampling and model errors.

➡ **Poterjoy (2022; QJRMS)** reveals the potential of blending LPF with EnKF to reduce the effects of sampling errors.

Motivation & Objectives

- Poterjoy (2022; QJRMS) introduces **regularization** and **tempering** steps for local PFs.

- **Kurosawa and Poterjoy (2023; MWR)** introduces an adaptive strategy for combining EnKF with the local PF.

➡ Data assimilation experiments using **WRF** with the blending filters.

The Local PF (Poterjoy 2022)

- The weight of the j^{th} state variable of the n^{th} particle is proportional to the likelihood:

$$w_j^n \propto p(y_i|\mathbf{x}_j^n)$$

y_i : i^{th} observation
$\mathbf{x}_j^n$: j^{th} state variable of the n^{th} particle

The Local PF (Poterjoy 2022)

- The weight of the j^{th} state variable of the n^{th} particle is proportional to the likelihood:

$$w_j^n \propto p(y_i|\mathbf{x}_j^n)$$

y_i : i^{th} observation
$\mathbf{x}_j^n$: j^{th} state variable of the n^{th} particle

- By defining $V_j^n = -\ln p(y_i|\mathbf{x}_j^n)$,

$$w_j^n \propto \exp(-V_j^n) \xrightarrow{\text{normalization}} w_j^n = \frac{\exp(-V_j^n)}{\sum \exp(-V_l^n)}$$

The Local PF (Poterjoy 2022)

- The weight of the j^{th} state variable of the n^{th} particle is proportional to the likelihood:

$$w_j^n \propto p(y_i|\mathbf{x}_j^n)$$

y_i : i^{th} observation
$\mathbf{x}_j^n$: j^{th} state variable of the n^{th} particle

- By defining $V_j^n = -\ln p(y_i|\mathbf{x}_j^n)$,

$$w_j^n \propto \exp(-V_j^n) \xrightarrow{\text{normalization}} w_j^n = \frac{\exp(-V_j^n)}{\sum \exp(-V_l^n)}$$

- Here, the **effective ensemble size** ($\boldsymbol{N}_{\mathbf{eff}}$; Liu and Chen 1998)

$$N_{\text{eff}} = \frac{1}{\sum (w_l^n)^2}$$

The Local PF (Poterjoy 2022)

- The weight of the j^{th} state variable of the n^{th} particle is proportional to the likelihood:

$$w_j^n \propto p(y_i|\mathbf{x}_j^n)$$

y_i : i^{th} observation
$\mathbf{x}_j^n$: j^{th} state variable of the n^{th} particle

- By defining $V_j^n = -\ln p(y_i|\mathbf{x}_j^n)$,

$$w_j^n \propto \exp(-V_j^n) \xrightarrow{\text{normalization}} w_j^n = \frac{\exp(-V_j^n)}{\sum \exp(-V_l^n)}$$

- Here, the **effective ensemble size** ($\boldsymbol{N}_{\mathbf{eff}}$; Liu and Chen 1998)

$$N_{\text{eff}} = \frac{1}{\sum (w_l^n)^2}$$

- To prevent weight collapse, a coefficient β_j is applied to adjust V_j^n, with the goal of achieving a **target effective ensemble size** (N_{eff}^t).

$$\beta_j = \arg\min_{\beta_j} \left(N_{\text{eff}}^t - \frac{1}{\sum (\hat{w}_l^n)^2} \right), \textit{ where } \hat{w}_j^n = \frac{\exp(-\beta_j V_j^n)}{\sum \exp(-\beta_l V_l^n)}$$

The Local PF (Poterjoy 2022)

- **Regularization** of particle weights provides one means of determining how to factor LPF into iterations, providing a natural framework for blending LPF with other filters.

- Mixing parameter $\boldsymbol{\kappa}$: N_x-dimensional vector $(0 \leq \kappa \leq 1)$

0: EnKF
1: LPF

$$\boldsymbol{\kappa} = \begin{bmatrix} \kappa^1 \\ \kappa^2 \\ \kappa^3 \\ \vdots \\ \kappa^{N_x} \end{bmatrix}$$

- Poterjoy (2022; QJRMS): **Uniform**
- Kurosawa and Poterjoy (2023; MWR): **Adaptive**

Adaptive Blending PF-Gaussian filters

Kurosawa and Poterjoy (2022; MWR)

Adaptive Blending PF-EnKF (Kurosawa and Poterjoy 2022)

Strategy:

Prior distribution is …

- Non-Gaussian: Iterative PF updates
- Gaussian: EnKF update

➡ **Shapiro-Wilk test**

Adaptive Blending PF-EnKF (Kurosawa and Poterjoy 2022)

1^{st} iteration

$$\boldsymbol{\kappa} = \begin{bmatrix} \text{NG} \\ \text{G} \\ \text{NG} \\ \vdots \\ \text{NG} \end{bmatrix}$$

Detected Gaussian by SWT

➡ No PF updates

N_x-dimensional vector

Adaptive Blending PF-EnKF (Kurosawa and Poterjoy 2022)

1st
iteration

$$\boldsymbol{\kappa} = \begin{bmatrix} \text{NG} \\ \text{G} \\ \text{NG} \\ \vdots \\ \text{NG} \end{bmatrix} = \begin{bmatrix} \\ 0.0 \\ \\ \vdots \\ \end{bmatrix} \qquad iter^2 = 0$$

Detected Gaussian by SWT
➡ No PF updates

N_x-dimensional vector

Adaptive Blending PF-EnKF (Kurosawa and Poterjoy 2022)

1st
iteration

$$\boldsymbol{\kappa} = \begin{bmatrix} 0.1 \\ 0.0 \\ 0.6 \\ \vdots \\ 0.1 \end{bmatrix} \quad \begin{bmatrix} \\ \\ \\ \\ \end{bmatrix} = \begin{bmatrix} 0.0 \\ \\ \vdots \\ \end{bmatrix} \qquad iter^2 = 0$$

β_1

N_x-dimensional vector

Adaptive Blending PF-EnKF (Kurosawa and Poterjoy 2022)

	1st iteration	2nd
	0.1	NG
	0.0	
$\boldsymbol{\kappa} =$	0.6	NG
	⋮	
	0.1	NG
	β_1	

$= \begin{bmatrix} \\ 0.0 \\ \\ \vdots \\ \\ \end{bmatrix}$ $iter^2 = 0$

N_x-dimensional vector

Adaptive Blending PF-EnKF (Kurosawa and Poterjoy 2022)

1st **2**nd
iteration

$$\boldsymbol{\kappa} = \begin{bmatrix} 0.1 + 0.2 \\ 0.0 \\ 0.6 + 0.1 \\ \vdots \\ 0.1 + 0.1 \end{bmatrix} = \begin{bmatrix} 0.0 \\ \\ \vdots \end{bmatrix} \quad iter^2 = 0$$

β_1 β_2

N_x-dimensional vector

Adaptive Blending PF-EnKF (Kurosawa and Poterjoy 2022)

1[st] iteration, **2**[nd], **3**[rd]

$$\boldsymbol{\kappa} = \begin{bmatrix} 0.1 + 0.2 & \text{NG} \\ 0.0 & \\ 0.6 + 0.1 & \text{G} \\ \vdots & \\ 0.1 + 0.1 & \text{NG} \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.7 \\ \vdots \end{bmatrix}$$

β_1 β_2

detected Gaussian by SWT

➡ No PF updates

$iter^2 = 0$

$iter^3 = 2$

N_x-dimensional vector

Adaptive Blending PF-EnKF (Kurosawa and Poterjoy 2022)

1st **2**nd **3**rd

iteration

$$\boldsymbol{\kappa} = \begin{bmatrix} 0.1 + 0.2 + 0.2 \\ 0.0 \\ 0.6 + 0.1 \\ \vdots \\ 0.1 + 0.1 + 0.2 \end{bmatrix} = \begin{bmatrix} \\ 0.0 \\ 0.7 \\ \vdots \\ \\ \end{bmatrix}$$

β_1 β_2 β_3

$iter^2 = 0$

$iter^3 = 2$

N_x-dimensional vector

Adaptive Blending PF-EnKF (Kurosawa and Poterjoy 2022)

1st iteration, **2nd**, **3rd**, **4th**

$$\boldsymbol{\kappa} = \begin{bmatrix} 0.1 + 0.2 + 0.2 & \text{G} \\ 0.0 & \\ 0.6 + 0.1 & \\ \vdots & \\ 0.1 + 0.1 + 0.2 & \text{NG} \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.0 \\ 0.7 \\ \vdots \end{bmatrix}$$

$\beta_1 \quad \beta_2 \quad \beta_3$

$iter^1 = 3$

$iter^2 = 0$

$iter^3 = 2$

N_x-dimensional vector

Adaptive Blending PF-EnKF (Kurosawa and Poterjoy 2022)

1st iteration, **2nd**, **3rd**, **4th**

$$\boldsymbol{\kappa} = \begin{bmatrix} 0.1 + 0.2 + 0.2 \\ 0.0 \\ 0.6 + 0.1 \\ \vdots \\ 0.1 + 0.1 + 0.2 + 0.2 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.0 \\ 0.7 \\ \vdots \end{bmatrix} \quad \begin{matrix} iter^1 = 3 \\ iter^2 = 0 \\ iter^3 = 2 \\ \end{matrix}$$

$\beta_1 \quad \beta_2 \quad \beta_3 \quad \beta_4$

$\boldsymbol{\kappa}$: N_x-dimensional vector

Adaptive Blending PF-EnKF (Kurosawa and Poterjoy 2022)

1st iteration, **2nd**, **3rd**, **4th**, **5th**

$$\boldsymbol{\kappa} = \begin{bmatrix} 0.1 + 0.2 + 0.2 & & & & \\ 0.0 & & & & \\ 0.6 + 0.1 & & & & \\ & & \vdots & & \\ 0.1 + 0.1 + 0.2 + 0.2 & & & & \text{NG} \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.0 \\ 0.7 \\ \vdots \end{bmatrix} \quad \begin{matrix} iter^1 = 3 \\ iter^2 = 0 \\ iter^3 = 2 \\ \\ \end{matrix}$$

β_1 β_2 β_3 β_4

N_x-dimensional vector

Adaptive Blending PF-EnKF (Kurosawa and Poterjoy 2022)

1st iteration, **2**nd, **3**rd, **4**th, **5**th

$$\boldsymbol{\kappa} = \begin{bmatrix} 0.1 + 0.2 + 0.2 \\ 0.0 \\ 0.6 + 0.1 \\ \vdots \\ 0.1 + 0.1 + 0.2 + 0.2 + 0.2 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.0 \\ 0.7 \\ \vdots \\ \end{bmatrix} \quad \begin{matrix} iter^1 = 3 \\ iter^2 = 0 \\ iter^3 = 2 \\ \\ \\ \end{matrix}$$

$\beta_1 \quad \beta_2 \quad \beta_3 \quad \beta_4 \quad \beta_5$

N_x-dimensional vector

Adaptive Blending PF-EnKF (Kurosawa and Poterjoy 2022)

1^{st} iteration **2^{nd}** **3^{rd}** **4^{th}** **5^{th}** **6^{th}**

$$\boldsymbol{\kappa} = \begin{bmatrix} 0.1 + 0.2 + 0.2 \\ 0.0 \\ 0.6 + 0.1 \\ \vdots \\ 0.1 + 0.1 + 0.2 + 0.2 + 0.2 \quad \text{NG} \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.0 \\ 0.7 \\ \vdots \end{bmatrix} \quad \begin{matrix} iter^1 = 3 \\ iter^2 = 0 \\ iter^3 = 2 \\ \\ \end{matrix}$$

β_1 β_2 β_3 β_4 β_5

N_x-dimensional vector

Adaptive Blending PF-EnKF (Kurosawa and Poterjoy 2022)

$$\boldsymbol{\kappa} = \begin{bmatrix} 0.1 + 0.2 + 0.2 \\ 0.0 \\ 0.6 + 0.1 \\ \vdots \\ 0.1 + 0.1 + 0.2 + 0.2 + 0.2 + 0.2 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.0 \\ 0.7 \\ \vdots \\ 1.0 \end{bmatrix} \quad \begin{matrix} iter^1 = 3 \\ iter^2 = 0 \\ iter^3 = 2 \\ \\ iter^n = 6 \end{matrix}$$

Columns (1st iteration, 2nd, 3rd, 4th, 5th, 6th): β_1, β_2, β_3, β_4, β_5, β_6

$\boldsymbol{\kappa}$: N_x-dimensional vector

Adaptive Blending PF-EnKF (Kurosawa and Poterjoy 2022)

1^{st} iteration, 2^{nd}, 3^{rd}, 4^{th}, 5^{th}, 6^{th}

$$\boldsymbol{\kappa} = \begin{bmatrix} 0.1 + 0.2 + 0.2 \\ 0.0 \\ 0.6 + 0.1 \\ \vdots \\ 0.1 + 0.1 + 0.2 + 0.2 + 0.2 + 0.2 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.0 \\ 0.7 \\ \vdots \\ 1.0 \end{bmatrix} \quad \begin{matrix} iter^1 = 3 \\ iter^2 = 0 \\ iter^3 = 2 \\ \\ iter^n = 6 \end{matrix}$$

$\beta_1 \quad \beta_2 \quad \beta_3 \quad \beta_4 \quad \beta_5 \quad \beta_6$

N_x-dimensional vector

no LPF update

no EnKF update

Adaptive Blending PF-EnKF (Kurosawa and Poterjoy 2022)

1^{st} iteration, 2^{nd}, 3^{rd}, 4^{th}, 5^{th}, 6^{th}

$$\boldsymbol{\kappa} = \begin{bmatrix} 0.1 + 0.2 + 0.2 \\ 0.0 \\ 0.6 + 0.1 \\ \vdots \\ 0.1 + 0.1 + 0.2 + 0.2 + 0.2 + 0.2 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.0 \\ 0.7 \\ \vdots \\ 1.0 \end{bmatrix} \quad \begin{matrix} iter^1 = 3 \\ iter^2 = 0 \\ iter^3 = 2 \\ \\ iter^n = 6 \end{matrix}$$

$\beta_1 \quad \beta_2 \quad \beta_3 \quad \beta_4 \quad \beta_5 \quad \beta_6$

no LPF update

no EnKF update

N_x-dimensional vector

$$\Rightarrow \boldsymbol{\eta} = 1 - \boldsymbol{\kappa} = \begin{bmatrix} 0.5 \\ 1.0 \\ 0.3 \\ \vdots \\ 0.0 \end{bmatrix}$$

NWP application

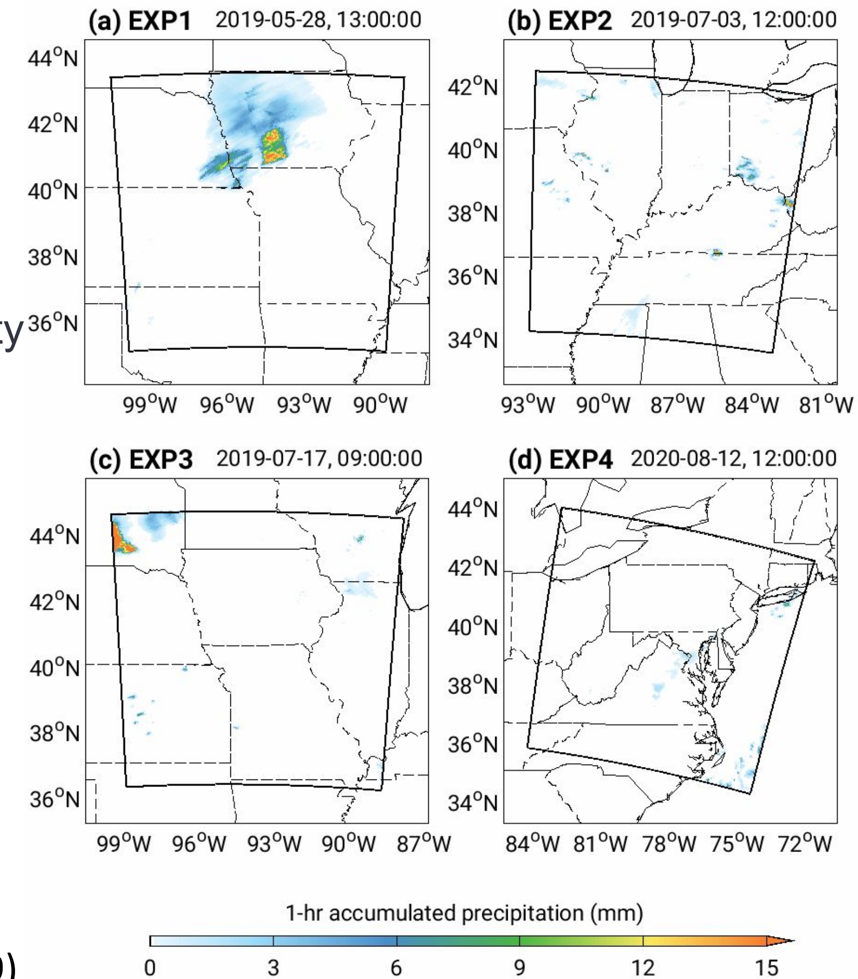
Experiments with WRF-DART

- **Numerical model** : **WRF**
  - 300 x 300 grid points x 50 layers (3-km grid spacing)
  - 15-min cycle
  - N_e: 64 mems

- **Observations** :
  - MADIS: METER, ACARS, routine soundings
  - NEXRAD: radar reflectivity and radial velocity
  - MRMS: clear-air reflectivity estimates

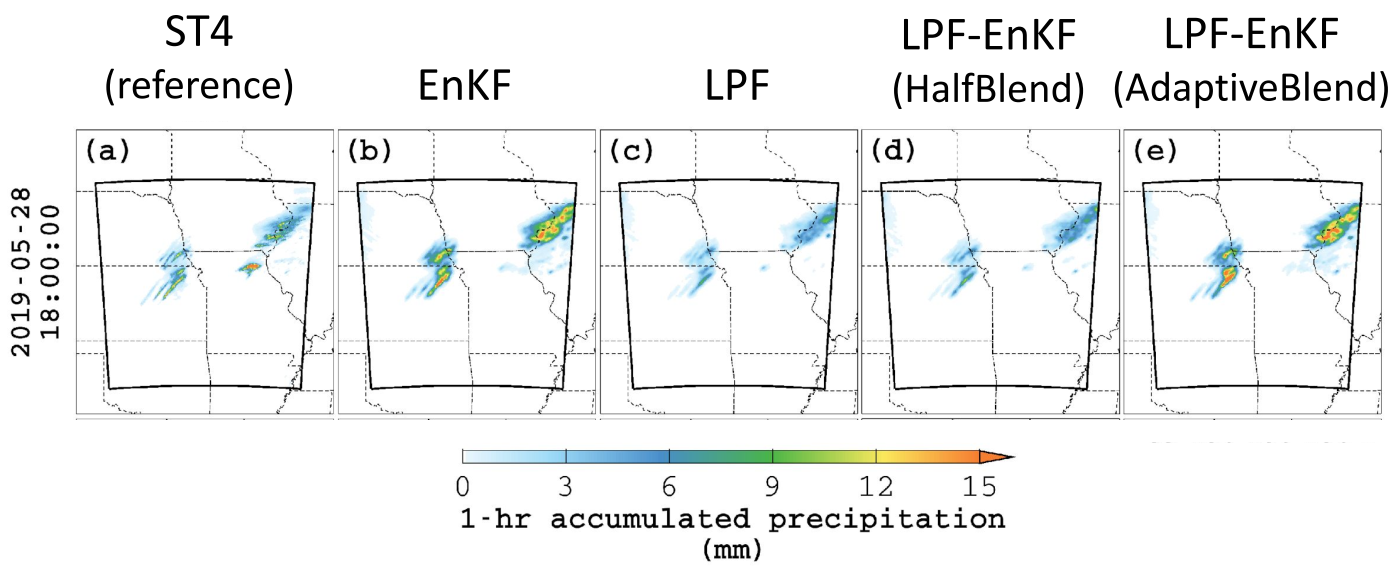
- **Data assimilation** :
  - EnKF
  - Iterative LPF (Poterjoy 2022)
  - LPF-EnKF-HalfBlend (50%-50%)
  - LPF-EnKF-AdaptiveBlend

  - Sensitivity experiments with and without the additive noise (Dowell and Wicker, 2009)



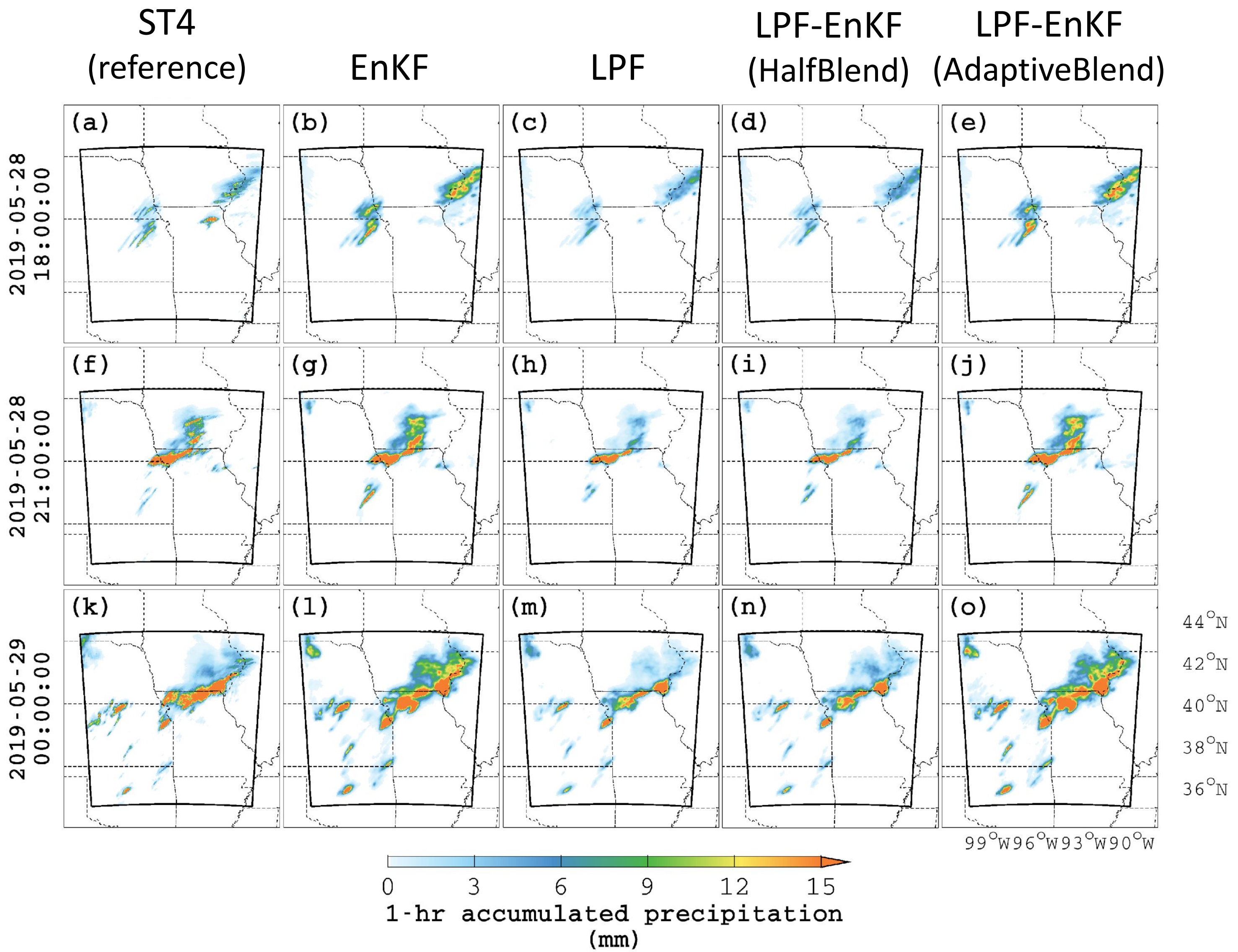
Experiments with WRF-DART

- Precipitation ($\overline{\boldsymbol{x}}^f$; **EXP1**)



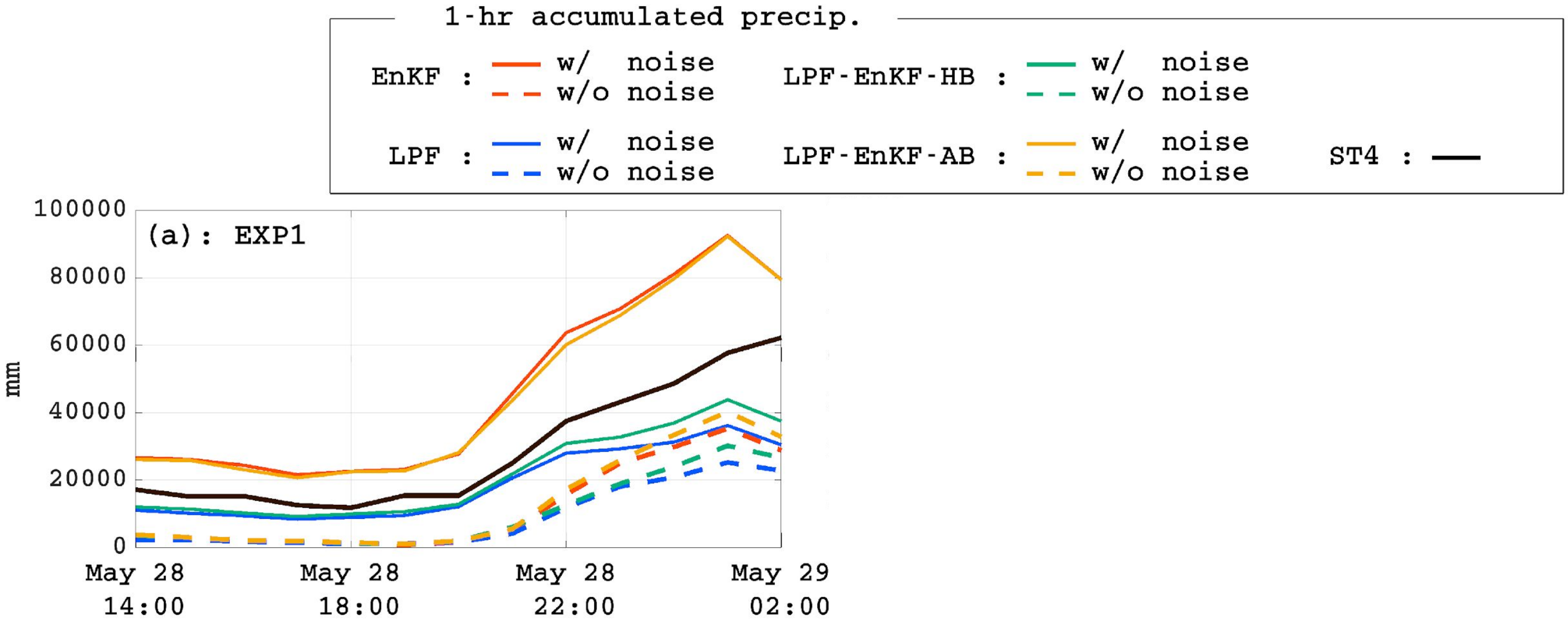
Experiments with WRF-DART

- Precipitation ($\overline{x}^f$; **EXP1**)



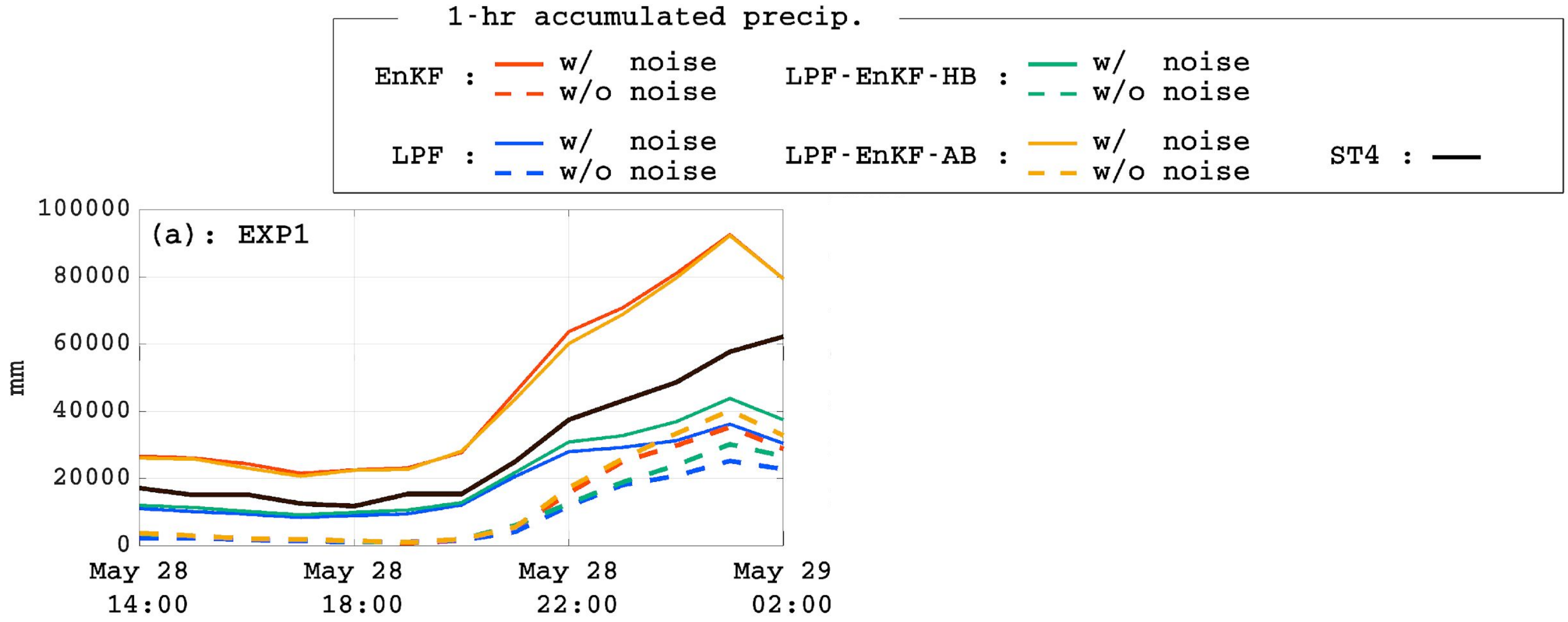
Experiments with WRF-DART

- Precipitation



Experiments with WRF-DART

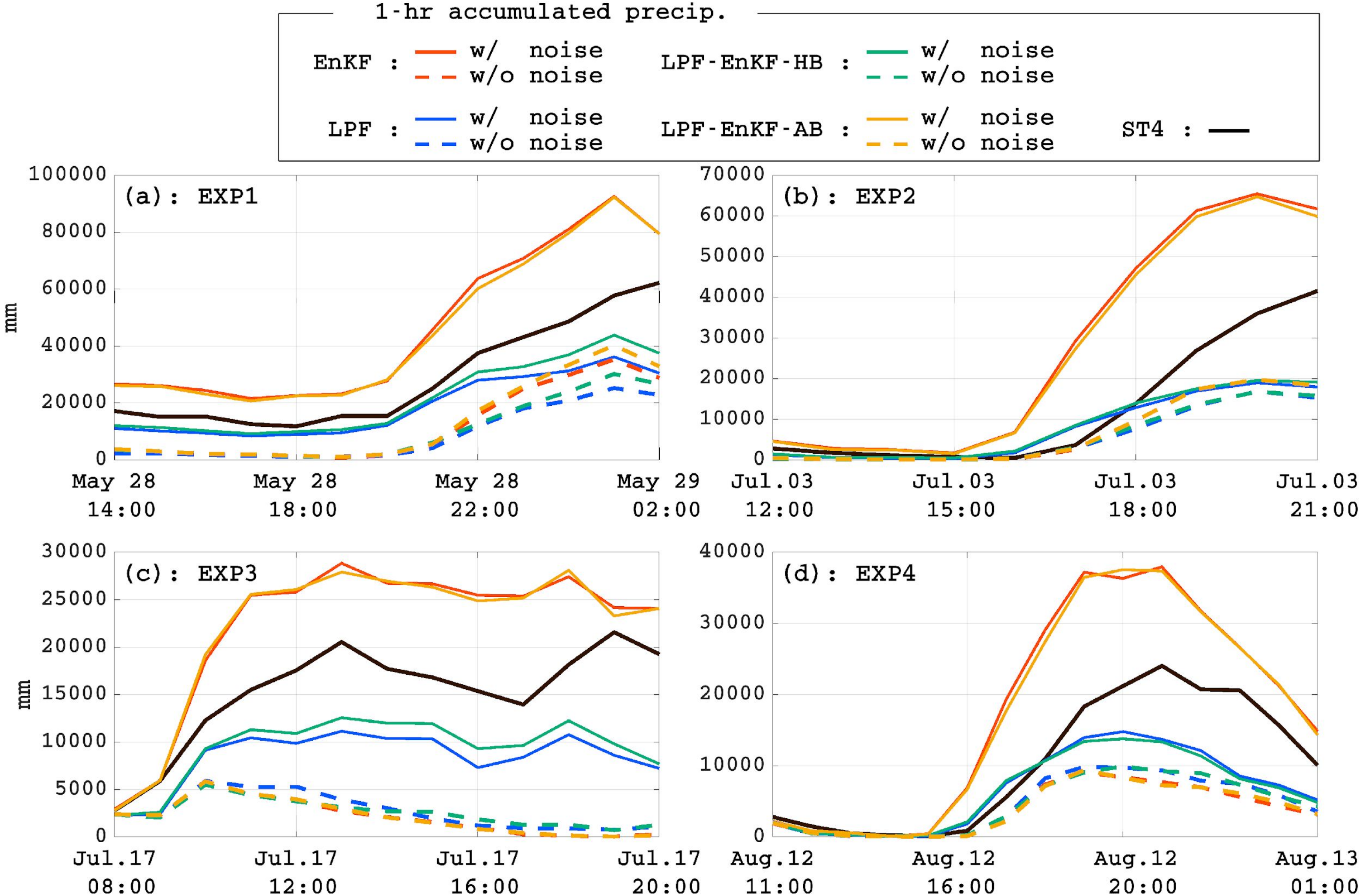
- Precipitation



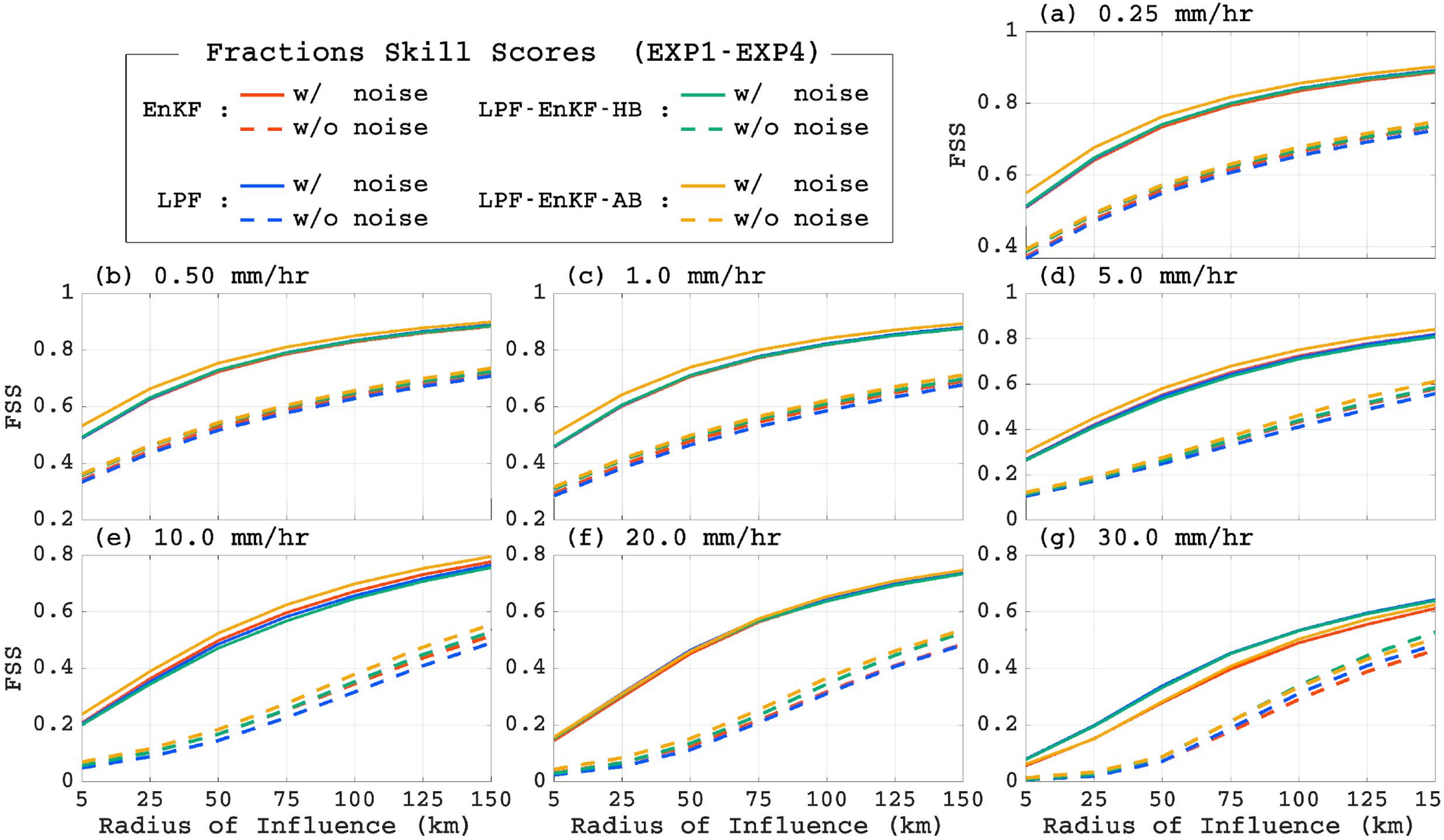
1. Additive inflation is necessary
2. EnKF displays a considerable increase with the additive noise
3. LPF does not exhibit as high of a sensitivity to the additive noise
4. Similarities between the adaptive blending and EnKF

Experiments with WRF-DART

- Precipitation



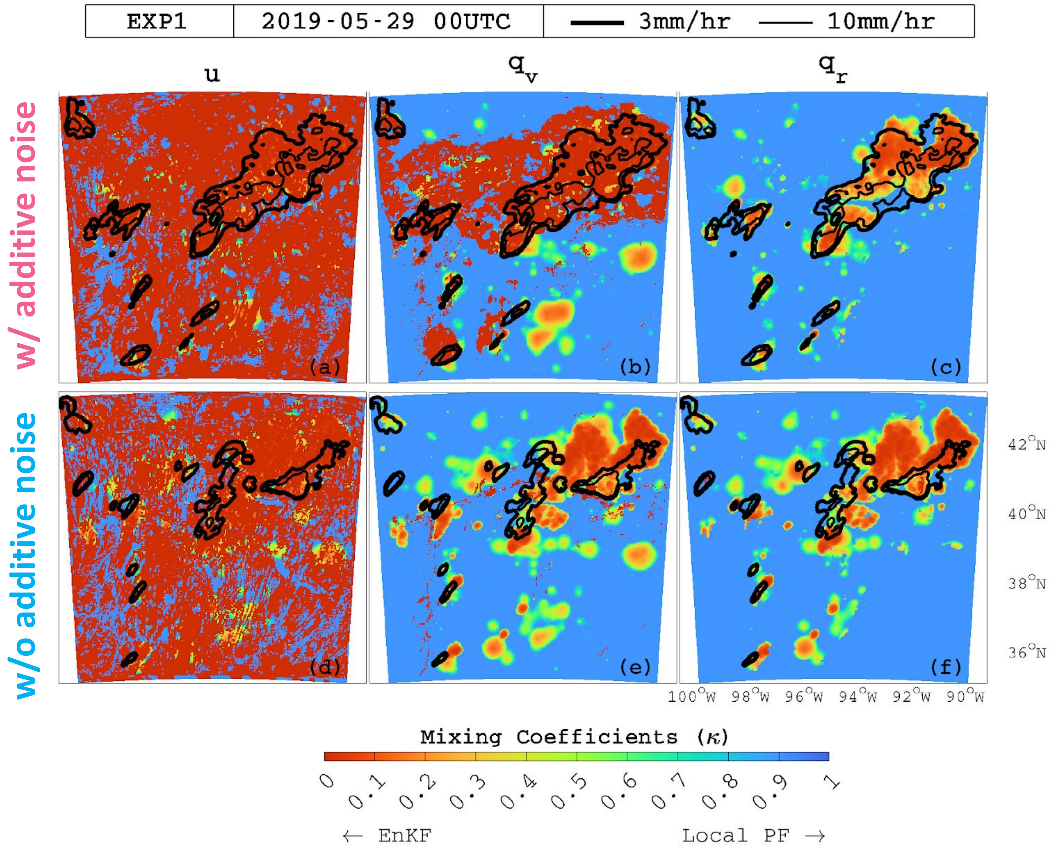
Experiments with WRF-DART



- Forecast verification with and without additive noise

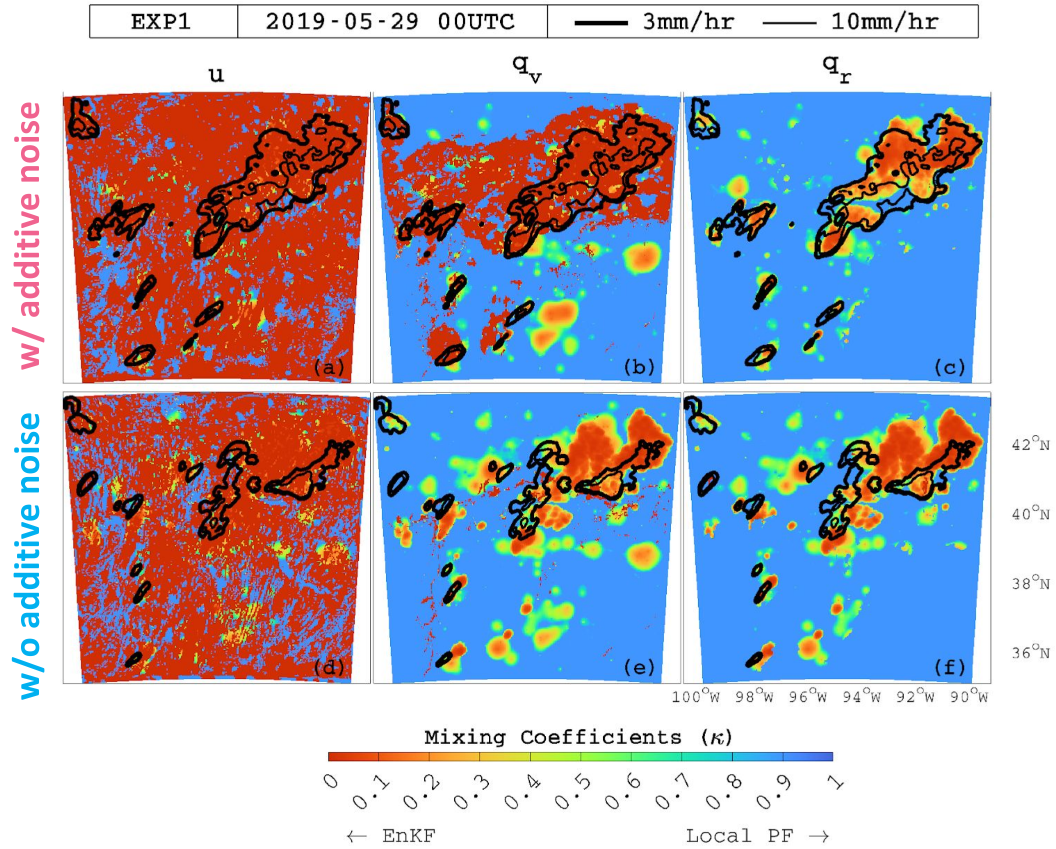
Experiments with WRF-DART

- Mixing parameter (κ; EXP1; the bottom layer)



Experiments with WRF-DART

- Mixing parameter (κ; EXP1; the bottom layer)



- **Center** of high precipitation: **Gaussian**
  - Persistently large amount of water vapor and liquid water across members
- **Outside** of high precipitation: **non-Gaussian**
  - Large diversity in members—many with low mixing ratios.

Summary

- Successfully implemented an adaptive LPF-EnKF for moist convection applications in WRF.

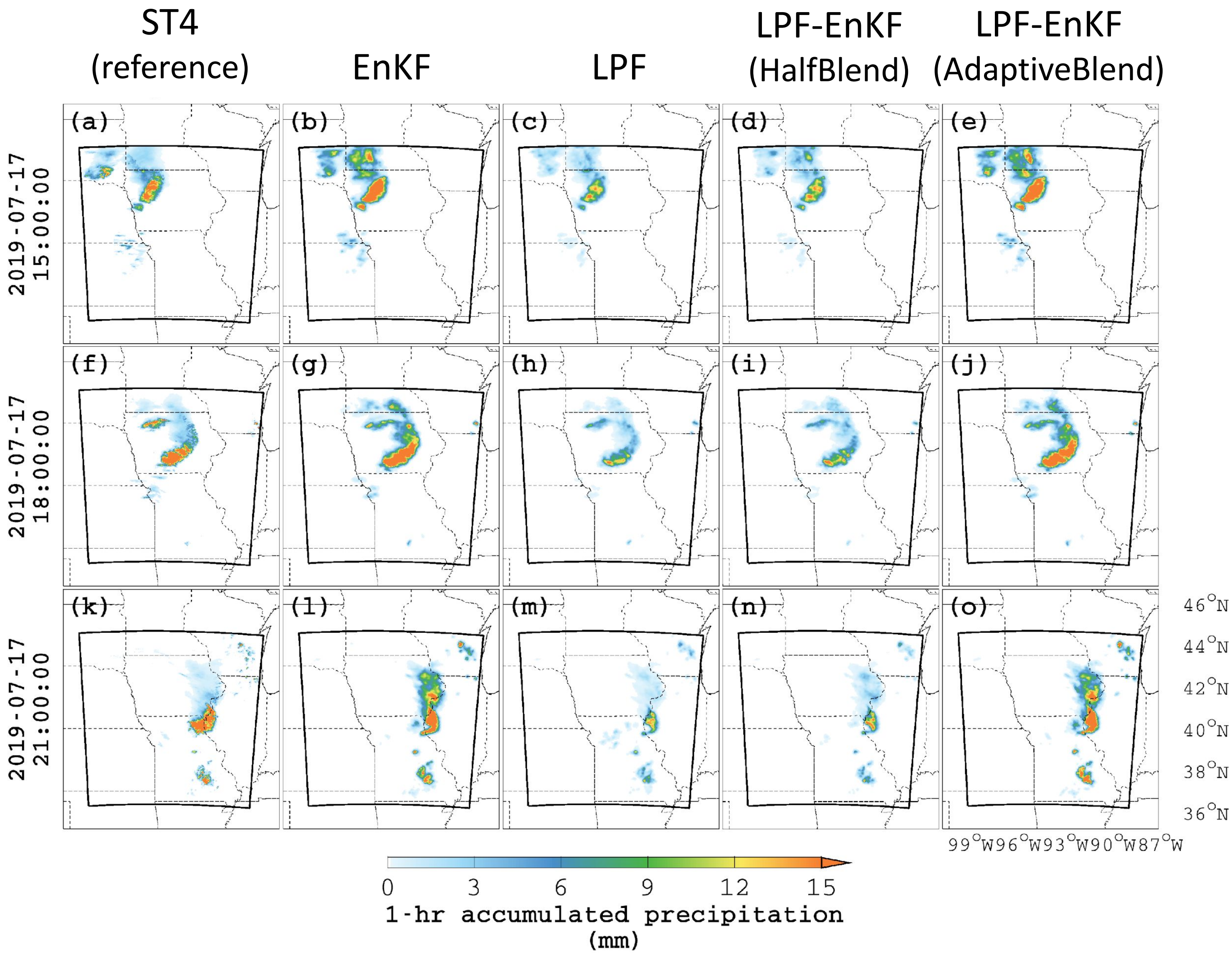
- The LPF and EnKF have varying sensitivities to additive inflation; EnKF is highly sensitive, while LPF is less so—which means adaptive filter is also sensitive to choices of inflation.

- Grid points with more precipitation are more frequently detected as Gaussian due to persistently similar dynamic, thermodynamic, and hydrometeor properties across members.

- Adaptive filter mostly appears to be effective at capturing "correct" filter for a given distribution.

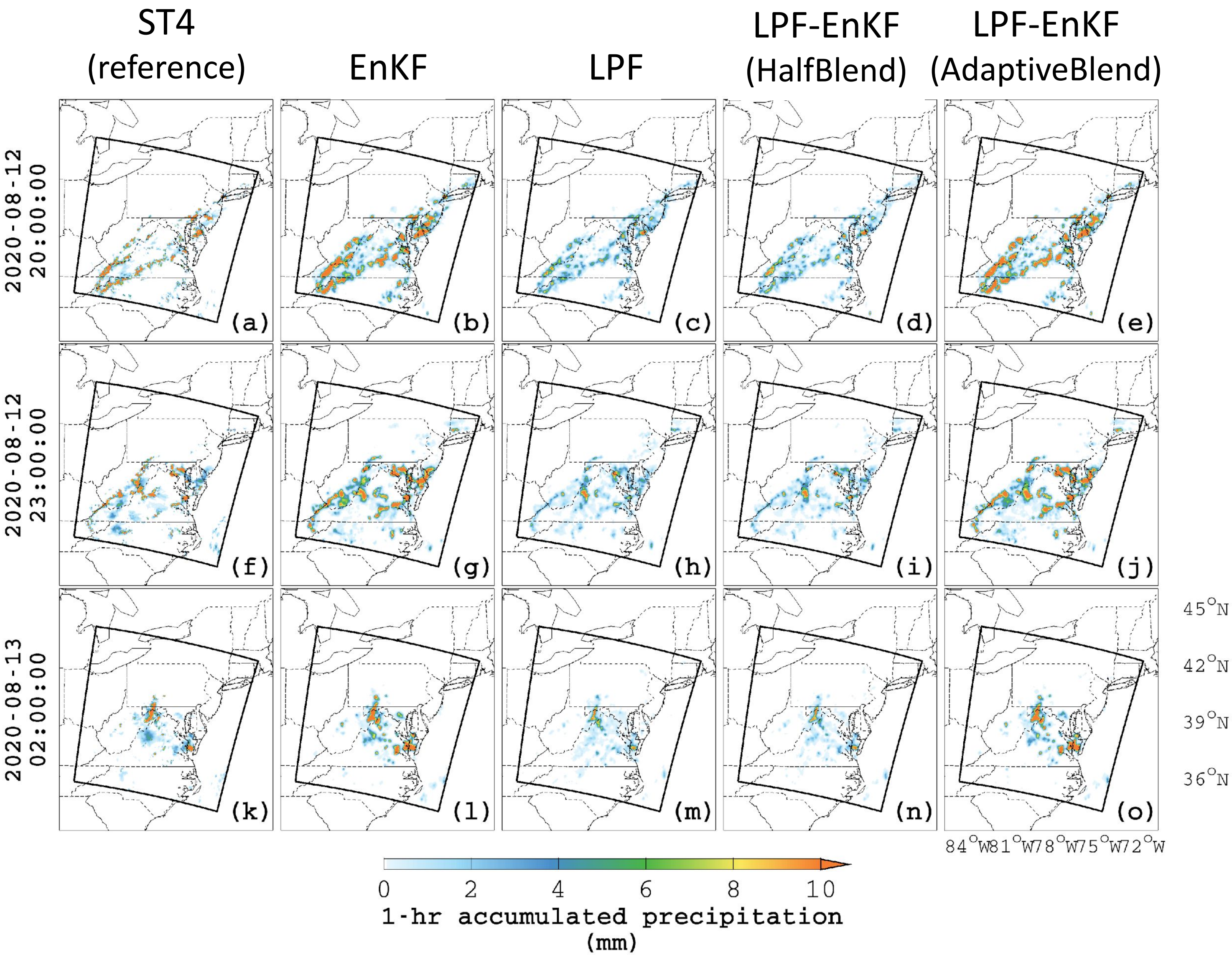
Experiments with WRF-DART

- Precipitation (**EXP3**)



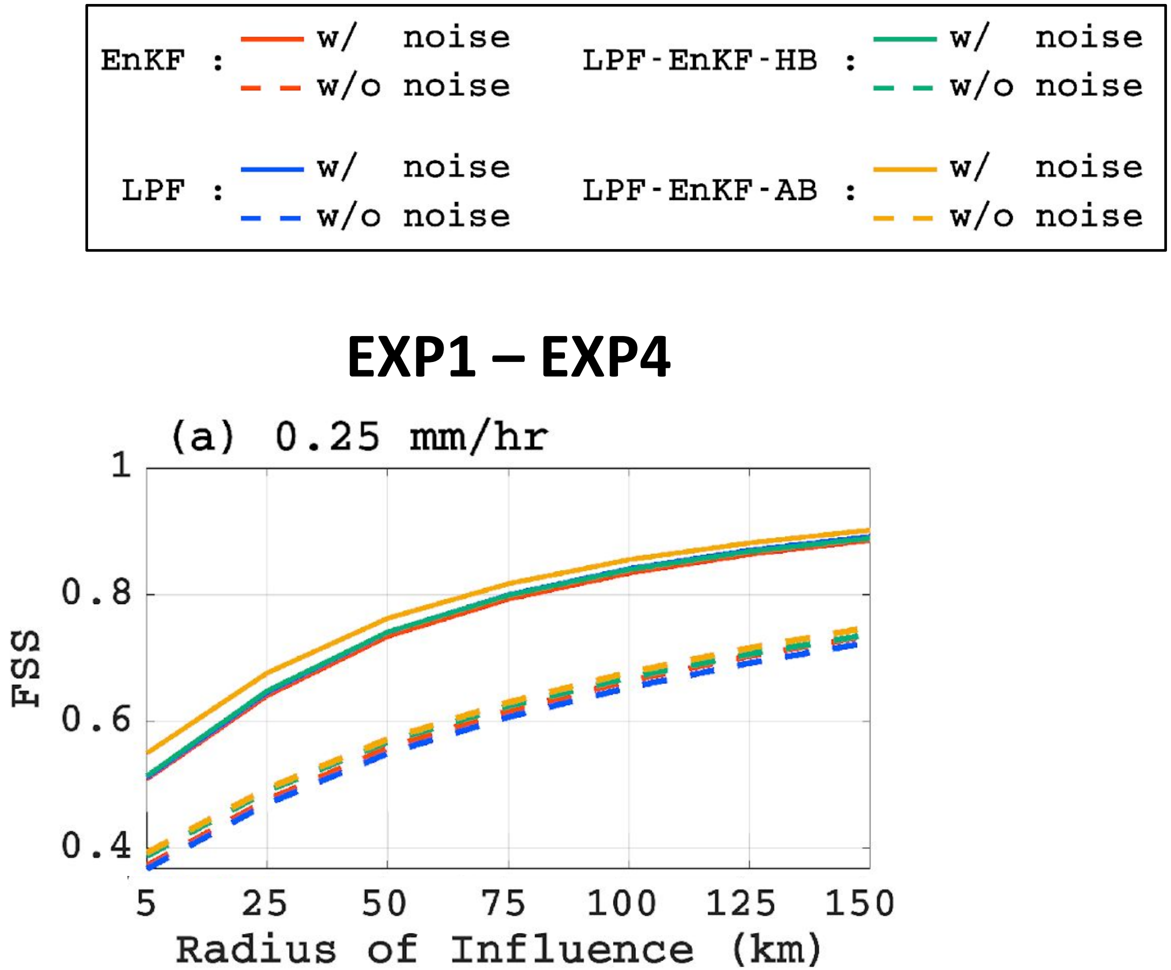
Experiments with WRF-DART

- Precipitation (**EXP4**)



Experiments with WRF-DART

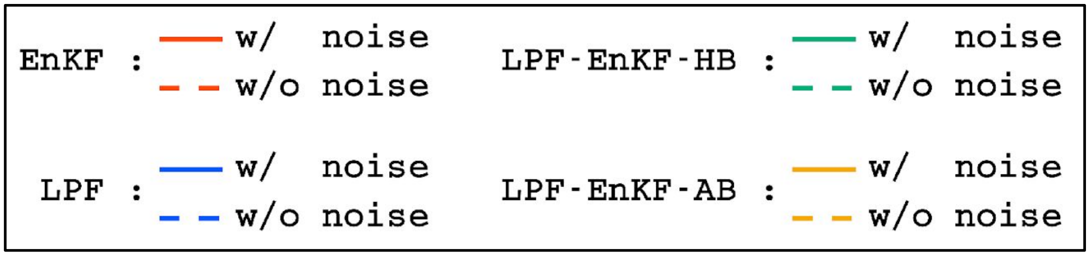
- Fractions Skill Scores



1: perfect forecast
0: no skill

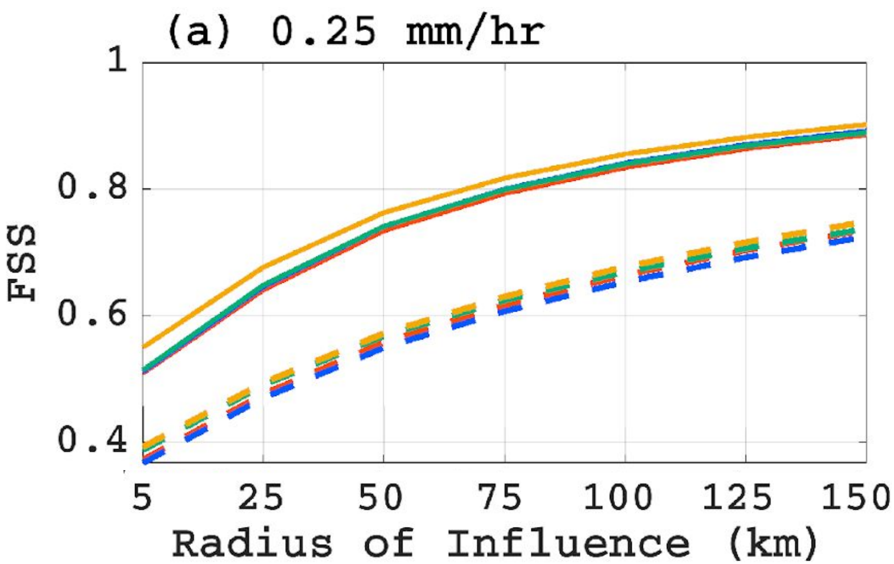
Experiments with WRF-DART

- Fractions Skill Scores

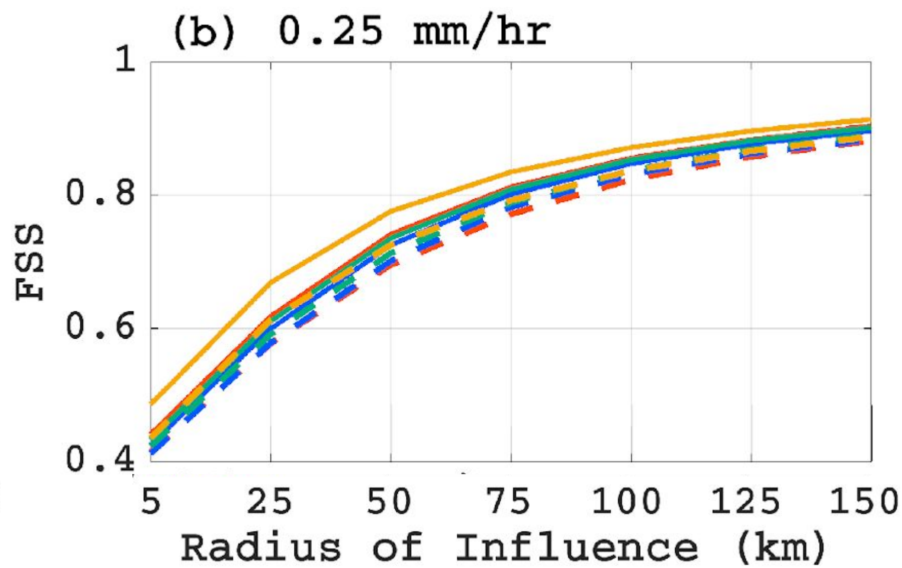


1: perfect forecast
0: no skill

EXP1 – EXP4



EXP4



Experiments with WRF-DART

- Mixing parameter (κ; **EXP2**)