

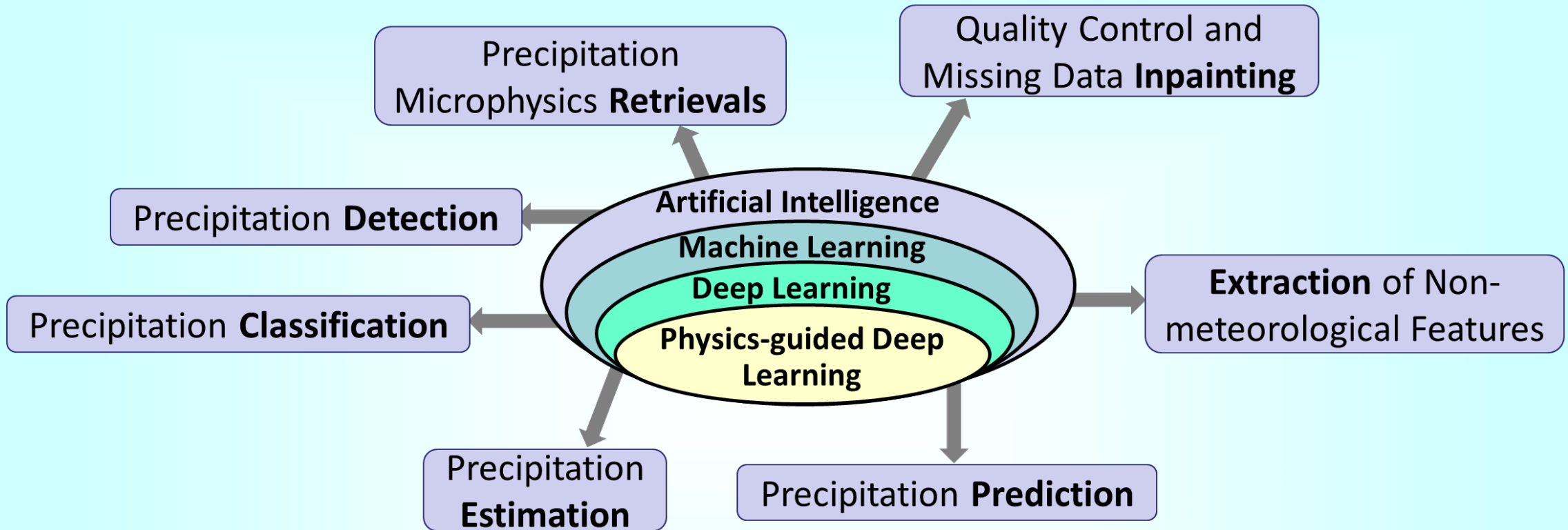
# Use of AI for Improving Radar Data Quality and Precipitation Estimates

**Haonan Chen**

Colorado State University, Fort Collins, CO

[haonan.chen@colostate.edu](mailto:haonan.chen@colostate.edu)

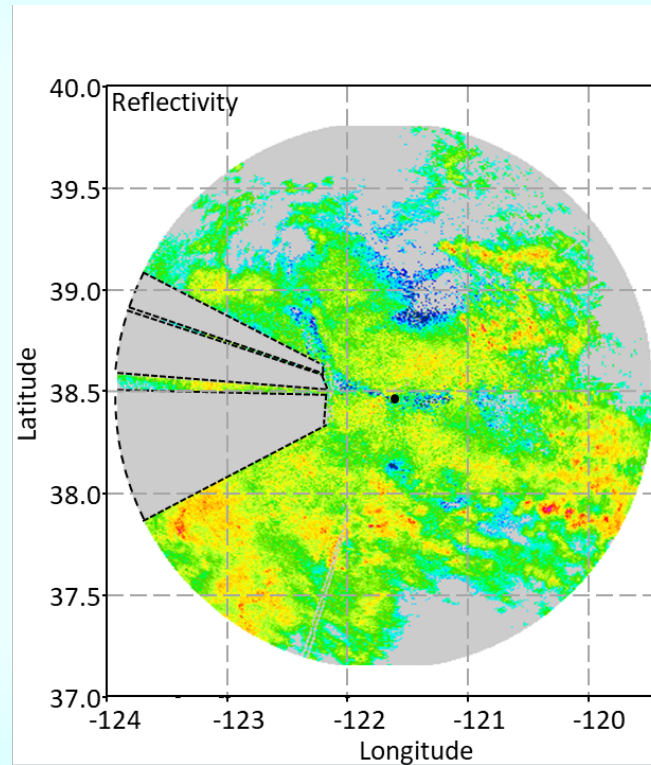
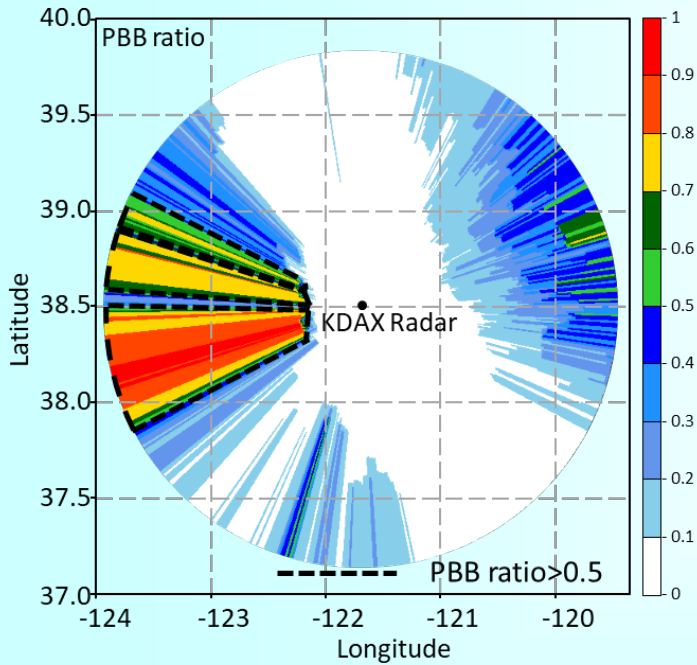
# Outline



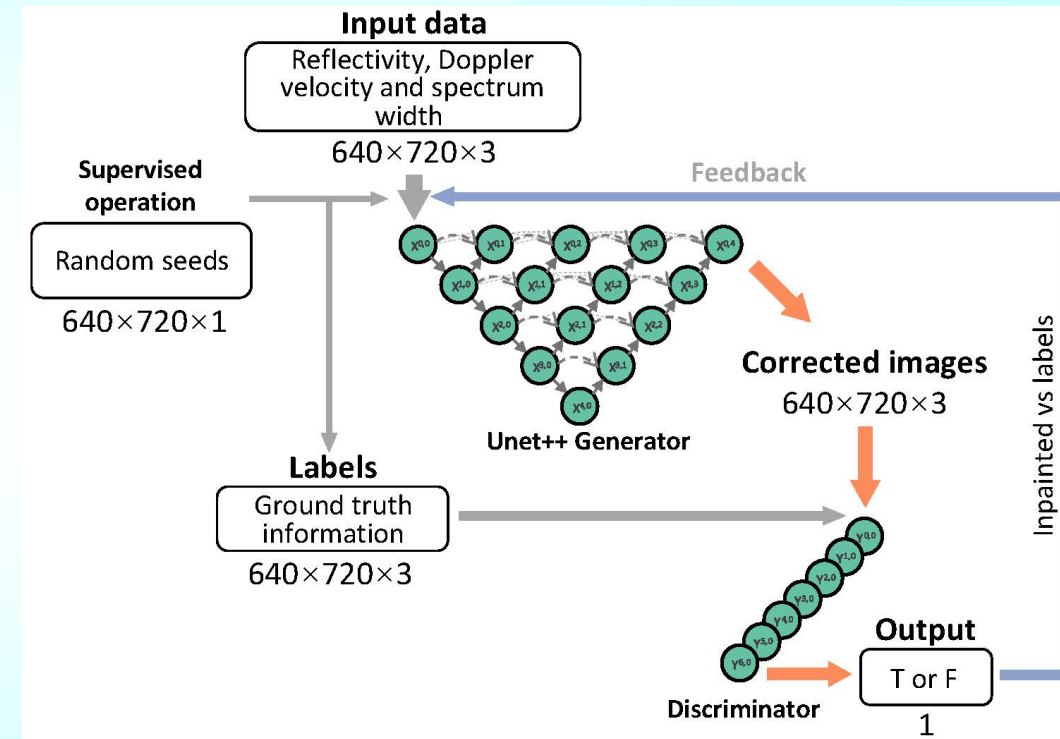
- AI/ML for radar beam blockage correction
- AI/ML for radar QPE

# AI/ML for Radar Beam Blockage Correction

## Beam blockages of the KDAX NEXRAD

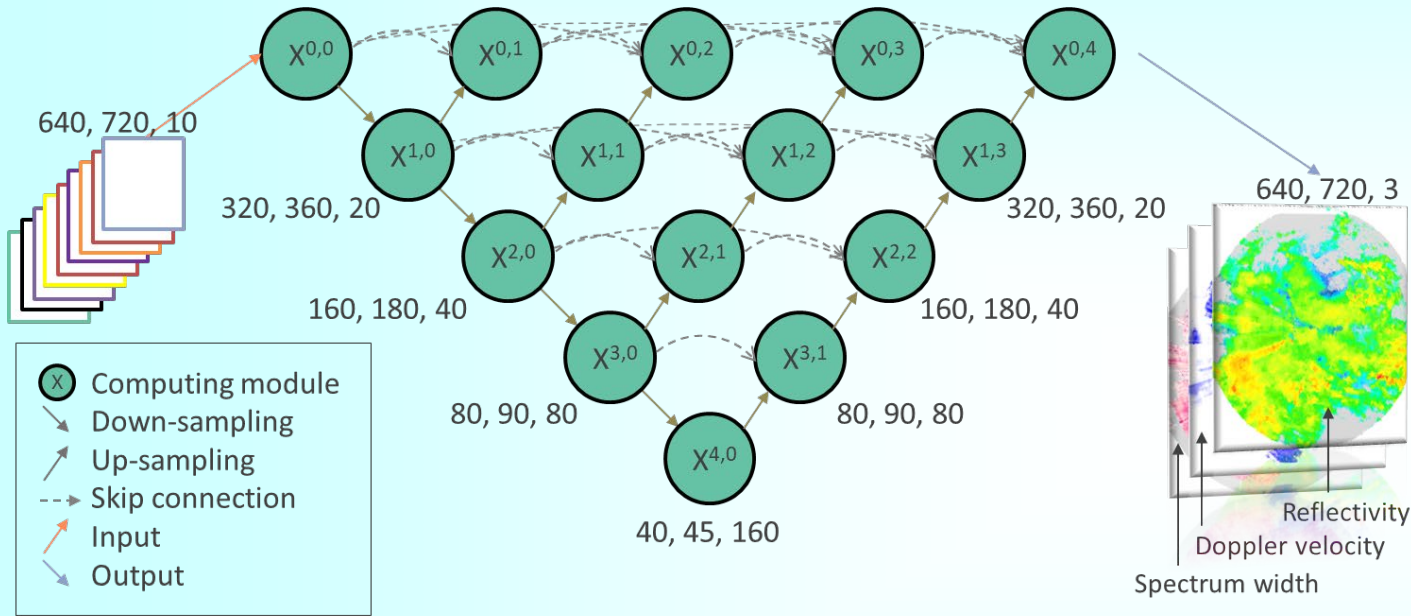


## Example AI/ML model for radar beam blockage correction



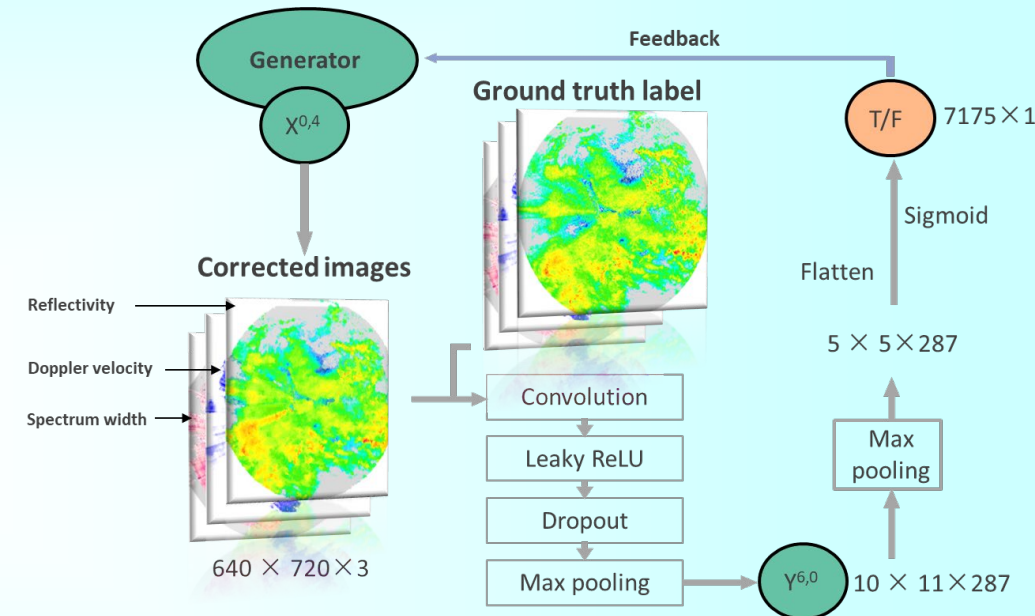
# AI/ML for Radar Beam Blockage Correction

## UNet++ based generator



- Down-sampling computing module:  $3 \times 3$  convolution with same padding + Leaky ReLU activation layer with a negative partial slope of 0.2 +  $2 \times 2$  max pooling
- Up-sampling computing module:  $2 \times 2$  deconvolution with a stride of 2+ concatenation layer +  $3 \times 3$  convolution + Leaky ReLU layer with a negative partial slope of 0.2
- Output layer: three-channel convolution with the sigmoid activation function

## Discriminator



Convolution filter:  $3 \times 3$   
Leaky ReLU layer slope: 0.2

Dropout rate: 0.1  
Max pooling filter:  $2 \times 2$

# AI/ML for Radar Beam Blockage Correction

## Model training

**Loss functions:**  $\mathcal{L}_G = \lambda E_{x,y,z}[|y - G(x, z)|] - E_{x,z} [\log(D(x, G(x, z)))]$  Generator loss function

$\mathcal{L}_D = -E_{x,y}[\log(D(x, y))] - E_{x,z} [\log(1 - D(x, G(x, z)))]$  Discriminator loss function

x: Input data (radar data generated by PBB mask)

y: Corresponding ground truth data (original radar data)

z: Random seeds input (Gaussian distribution random seeds)

G: Generator prediction

D: Discriminator output

$\lambda$ : To regulate the weight distribution between the two components

## Training parameters:

RxA: Range and azimuth size of input.

$C_0$ : Number of channels after multi-channel convolution.

g: Multiple after each down-sampling.

Depth: Layer numbers of Unet++.

Batch Size: Number of samples per batch.

Training iterations: Total number of iterations trained.

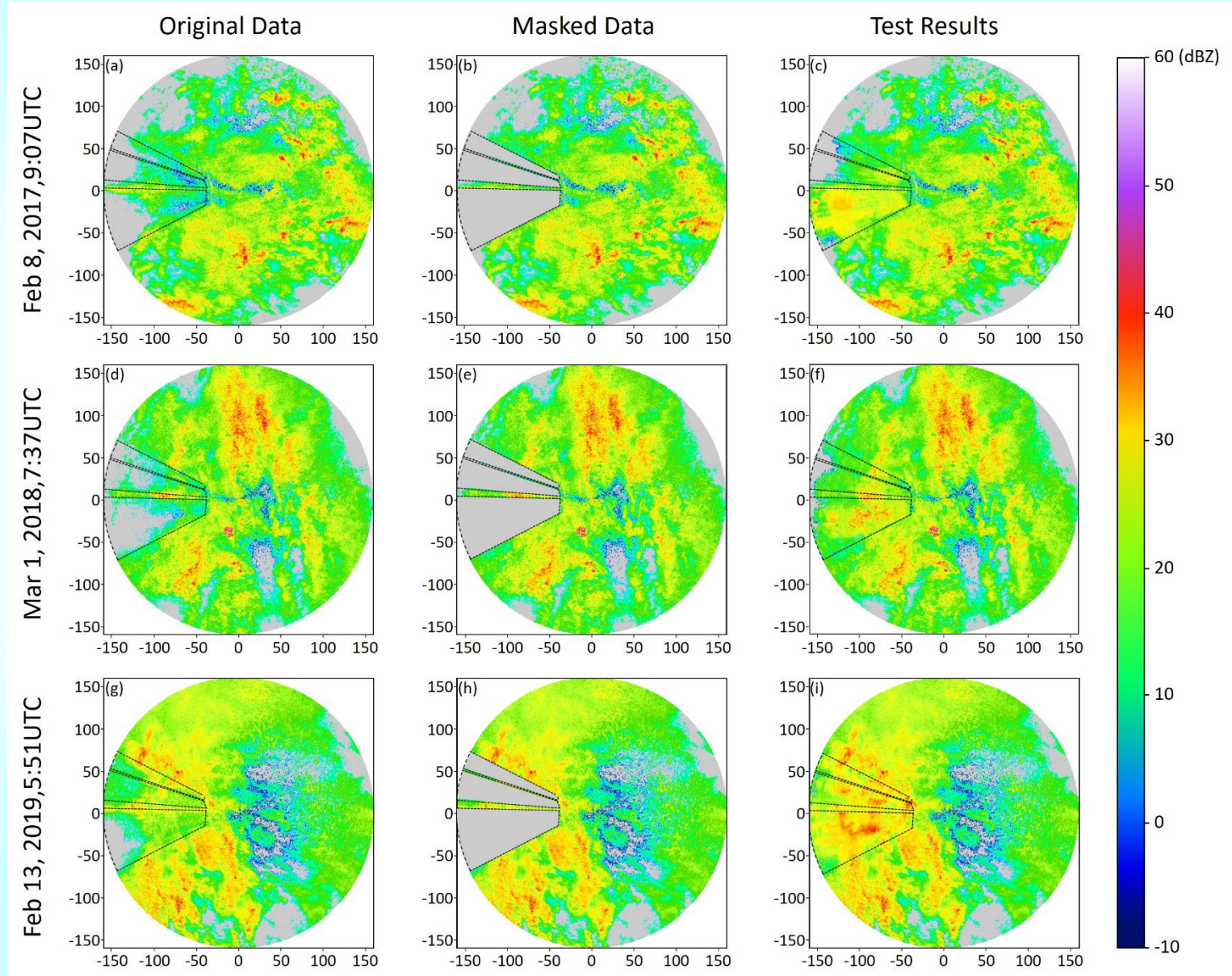
LR-Reduction: Batch number when learning rate is reduced with  $\beta_1$  equals to 0.5.

Model	RxA	$C_0$	g	Depth	Batch Size	Iterations	LR-Reduction
<b>KDAX</b>	640×720	10	2	5	16	$2 \times 10^4$	$1.9 \times 10^4$

- Time range: November to February, 2019-2021
- Radar observation size: 640×720 (range: 160km; azimuth: 0-360°)
- 80% for training, 20% for validation

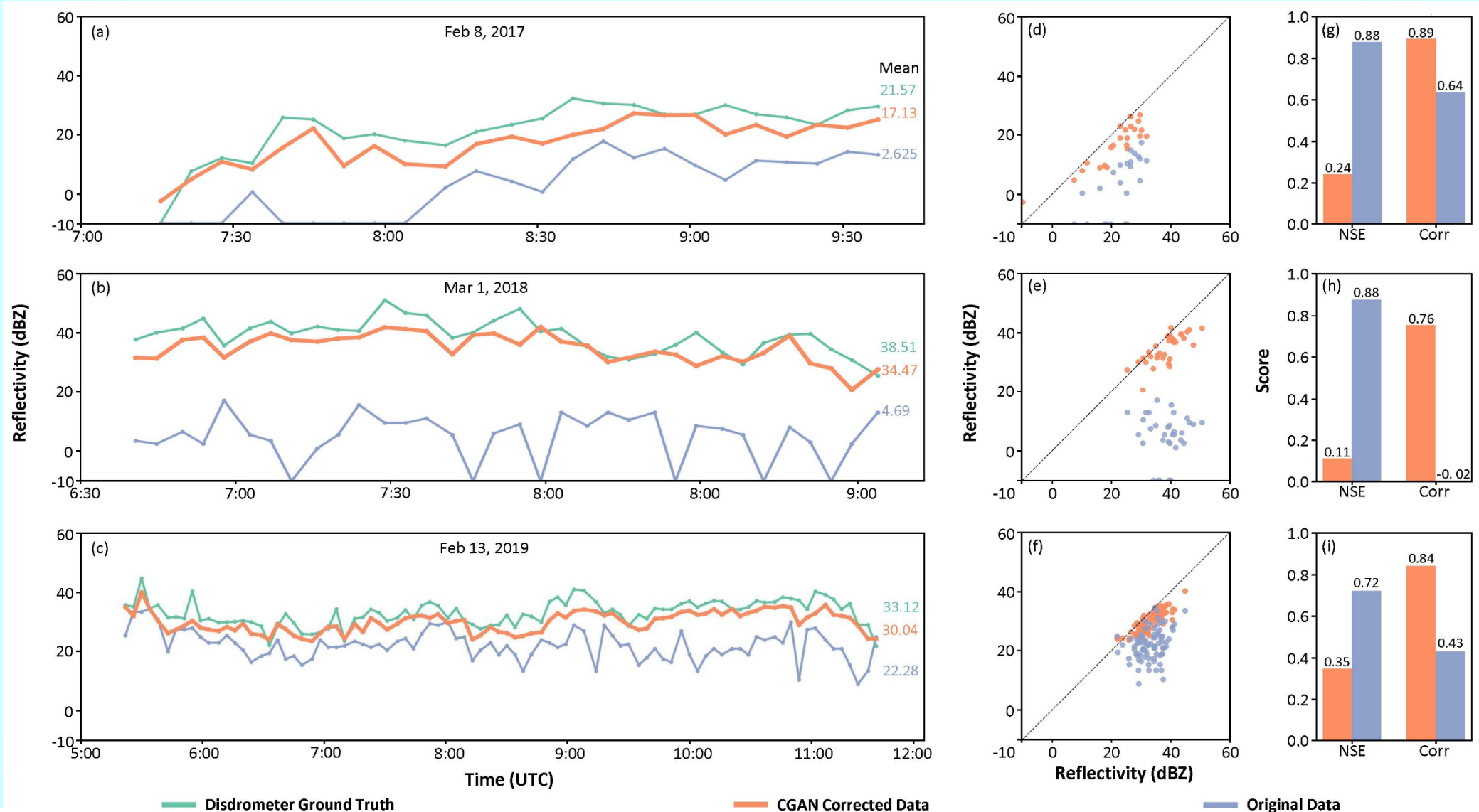
# AI/ML for Radar Beam Blockage Correction

## Testing results and evaluation



# AI/ML for Radar Beam Blockage Correction

## Testing results and evaluation



# AI for Radar Quantitative Precipitation Estimation

## Traditional radar rainfall relations

$$R(Z_h) = aZ_h^b$$

$$R(Z_h, Z_{dr}) = aZ_h^b Z_{dr}^c$$

$$R(K_{dp}) = aK_{dp}^b$$

$$R(Z_{dr}, K_{dp}) = aZ_{dr}^b K_{dp}^c$$

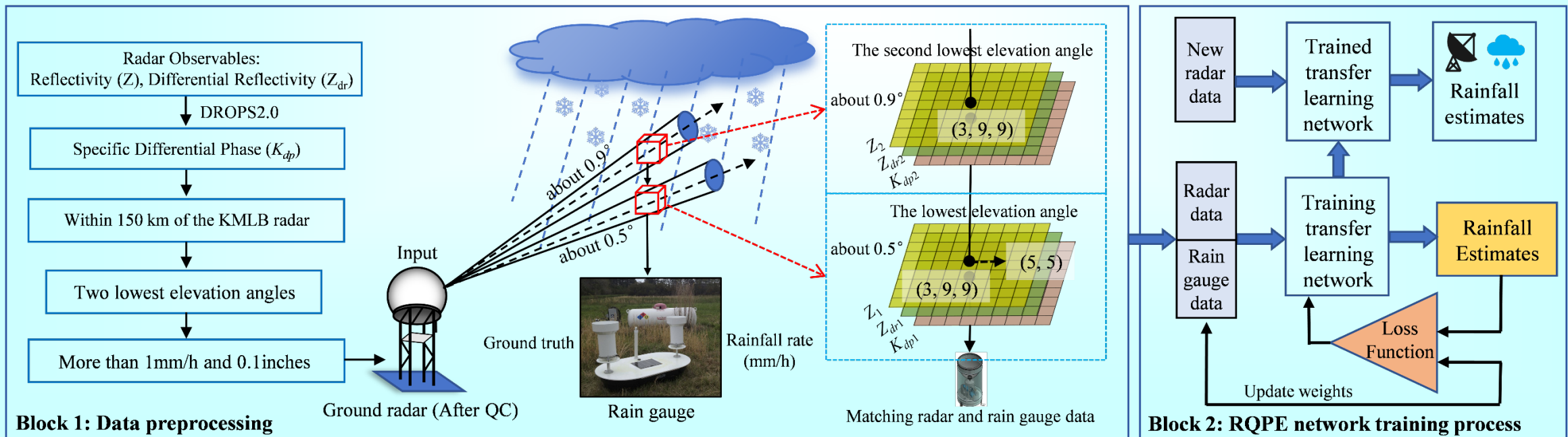
$$R(A) = aA^b$$

**Problem 1:** Parameterization error is high.

**Problem 2:** Generalization capability of these relations is low, e.g., the relations derived for deep convective events may not work well for shallow/warm cloud systems.

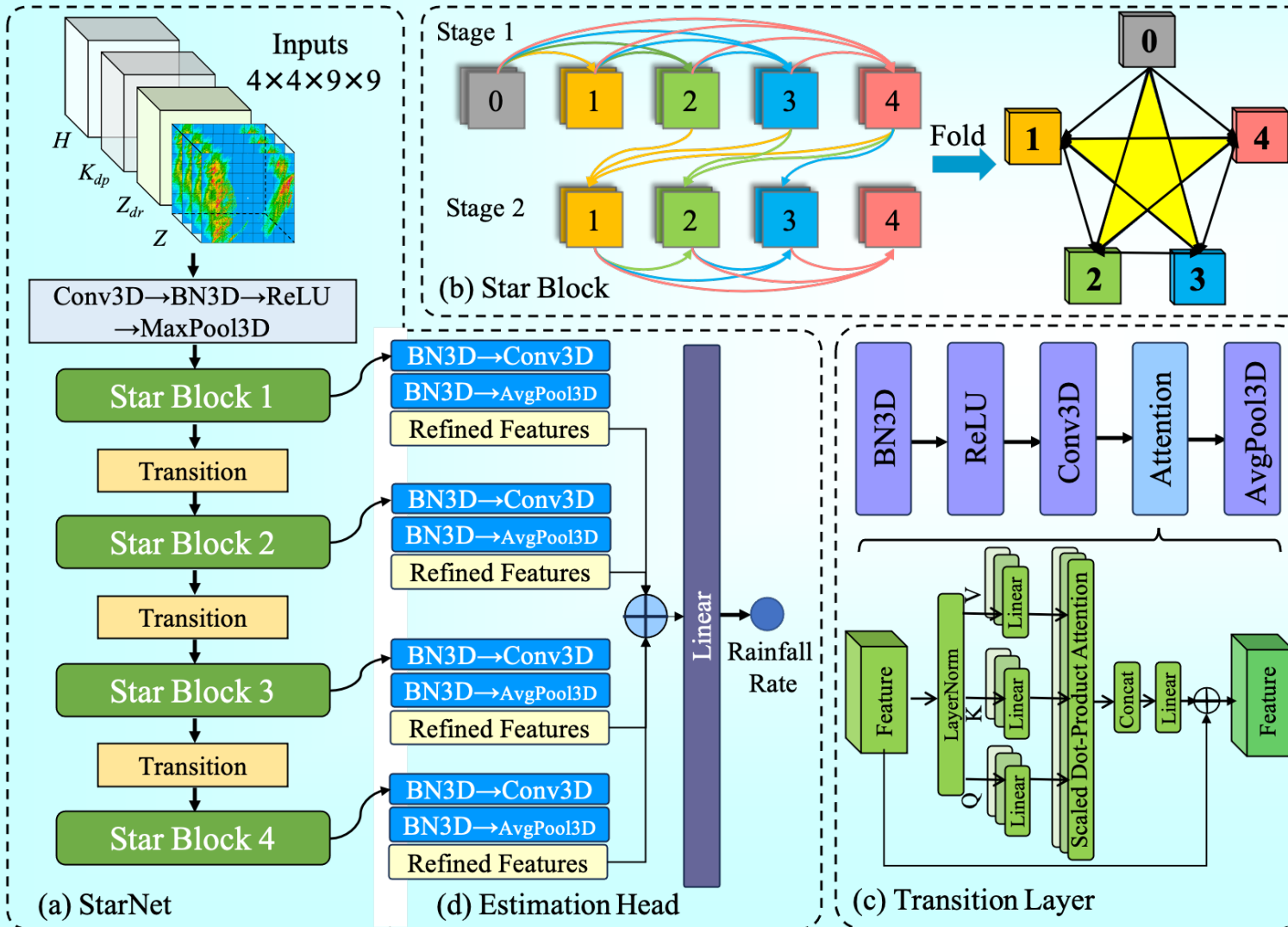
**Problem 3 (Ground-based radars):** Radar measurements provide indirect mapping of surface QPE.

## AI-based radar QPE



# AI for Radar Quantitative Precipitation Estimation

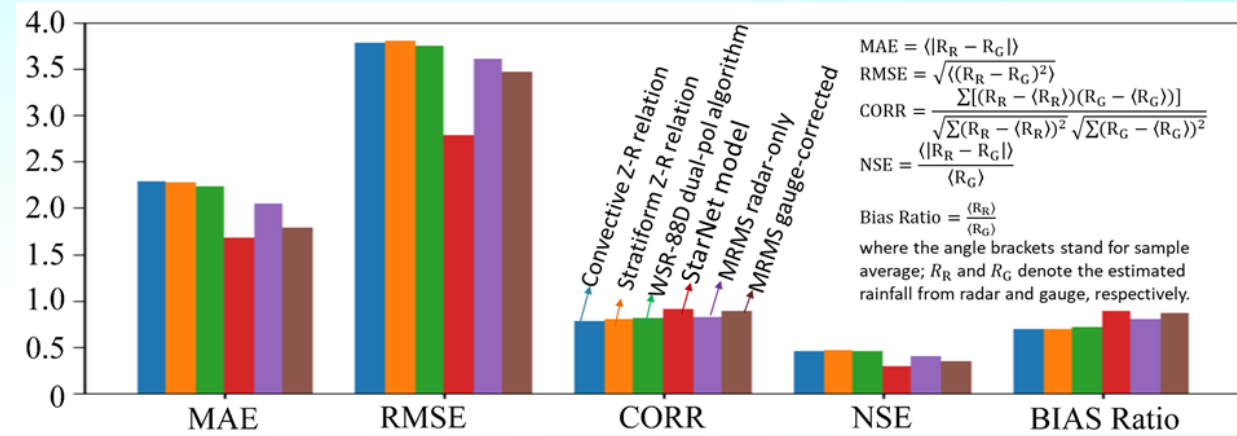
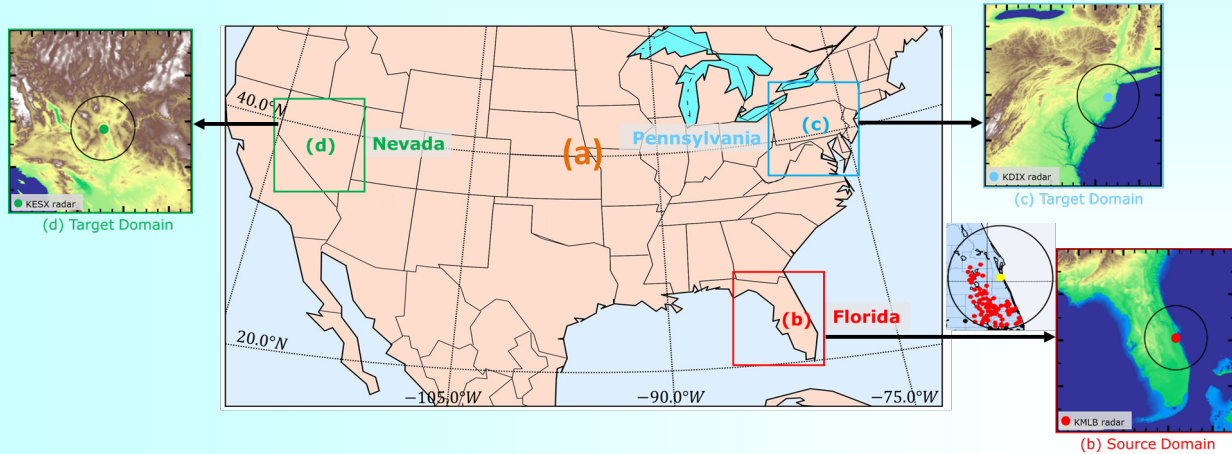
## QPE based on multidimensional multiparameter radar observations (StarNet model)



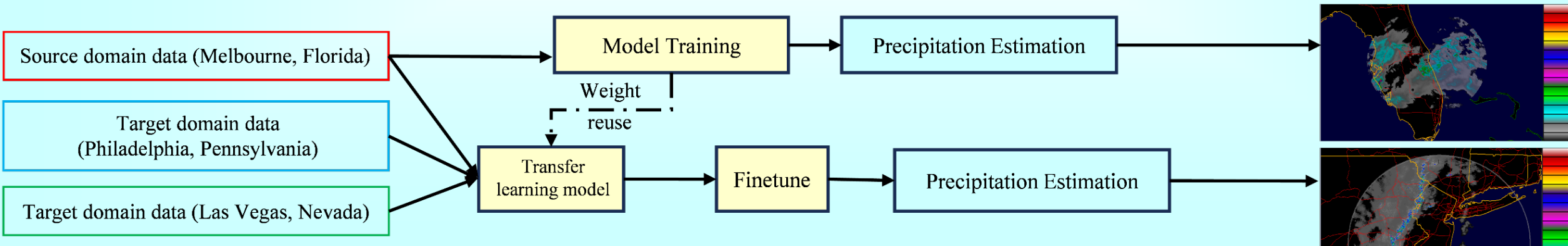
- Each star block incorporates a circular connection among its layers, enhancing inter-layer feature refinement. Each block has six cascaded bottleneck layers.
- Stage 1 focuses on the flow of radar observation features from shallow to deep features, while stage 2 reverses this flow, facilitating the interaction and fusion of features across levels.
- The transition layers are used to downsample features and filter key features.
- The estimation head module integrates features at different semantic levels.

# AI for Radar Quantitative Precipitation Estimation

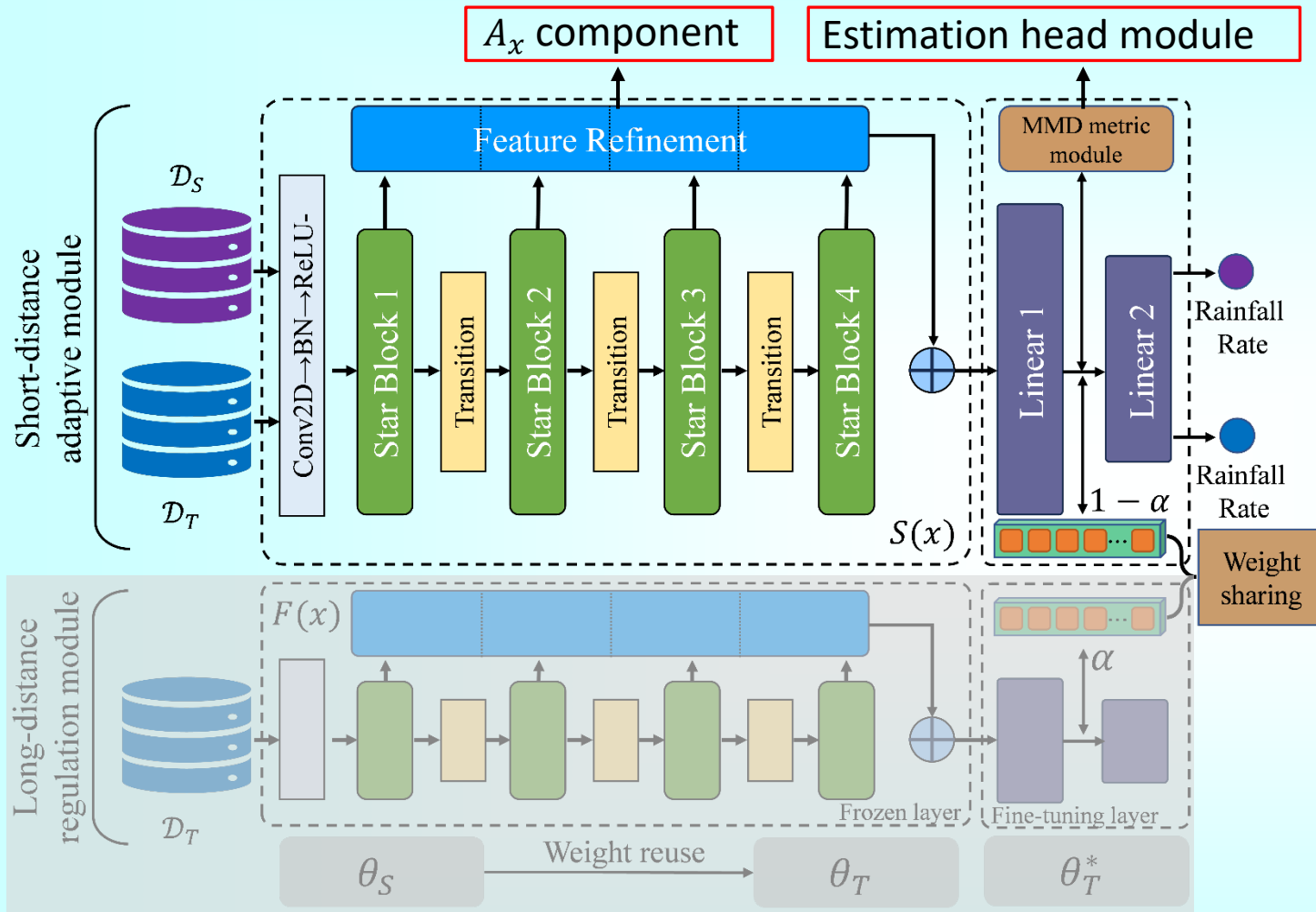
## Baseline model performance in the source domain (KMLB radar in FL)



- How to use the knowledge learned from a source domain in target domains?
- How to train and apply an AI/ML model when limited observations are available for model training?



# Transfer Learning to Improve AI QPE Model Generalization



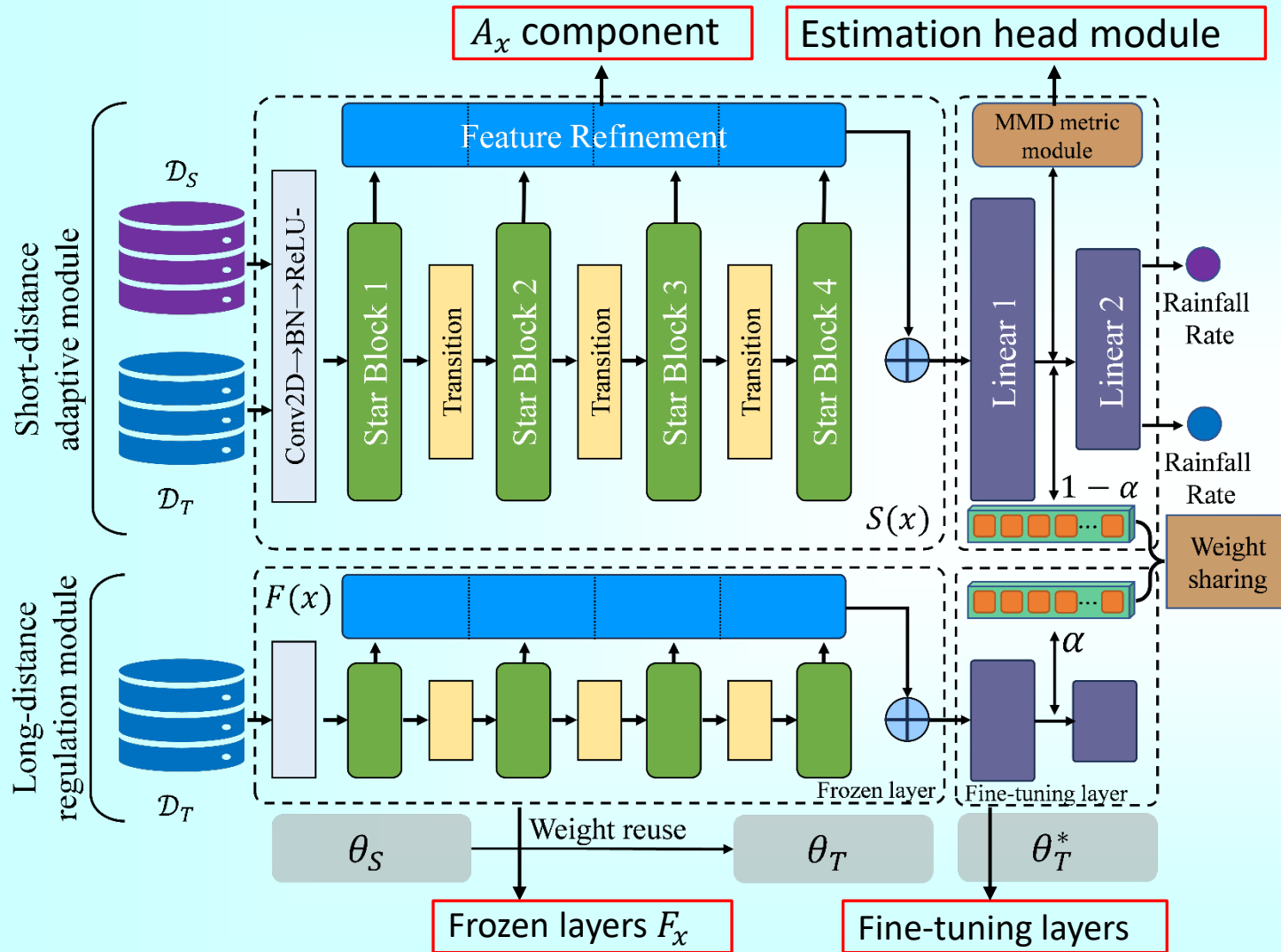
## • A short-distance adaptation module and a long-distance adjustment module.

- The short-distance adaptation module enhances the model's generalization capability when data distributions are similar.
- It is structured as a dual-path weight-sharing network. The first path processes data from the source domain, while the second path handles data from the target domains. This module comprises an  $A_x$  component and a precipitation estimation head module.
- Only the precipitation estimation head module will be fine-tuned, which has two fully connected layers, an adaptive layer with weight  $1-\alpha$ , and a maximum mean discrepancy (MMD) layer.

$$MMD(D_S, D_T) = \left\| \frac{1}{N_S} \sum_{i=1}^{N_S} \phi(x_i^S) - \frac{1}{N_T} \sum_{j=1}^{N_T} \phi(x_j^T) \right\|_H$$

where  $\phi$  is the feature mapping space, and  $H$  is the reproducing kernel Hilbert space. A small MMD indicates high similarity between source and target domain feature distributions, allowing direct transfer learning; otherwise, methods to reduce this difference are necessary.

# Transfer Learning to Improve AI QPE Model Generalization



- When the difference on data distribution is substantial, we employ an  $\alpha$  fusion algorithm, formulated as follows:

$$F(x) \oplus A(x) = \alpha F(x) + (1 - \alpha) A(x)$$

where  $\alpha$  ranges from 0 to 1, denoting the degree to which information from the source domain is preserved or the information from the target domain to be incorporated.  $\alpha$  decrease will make the model closer to the Finetune approach.

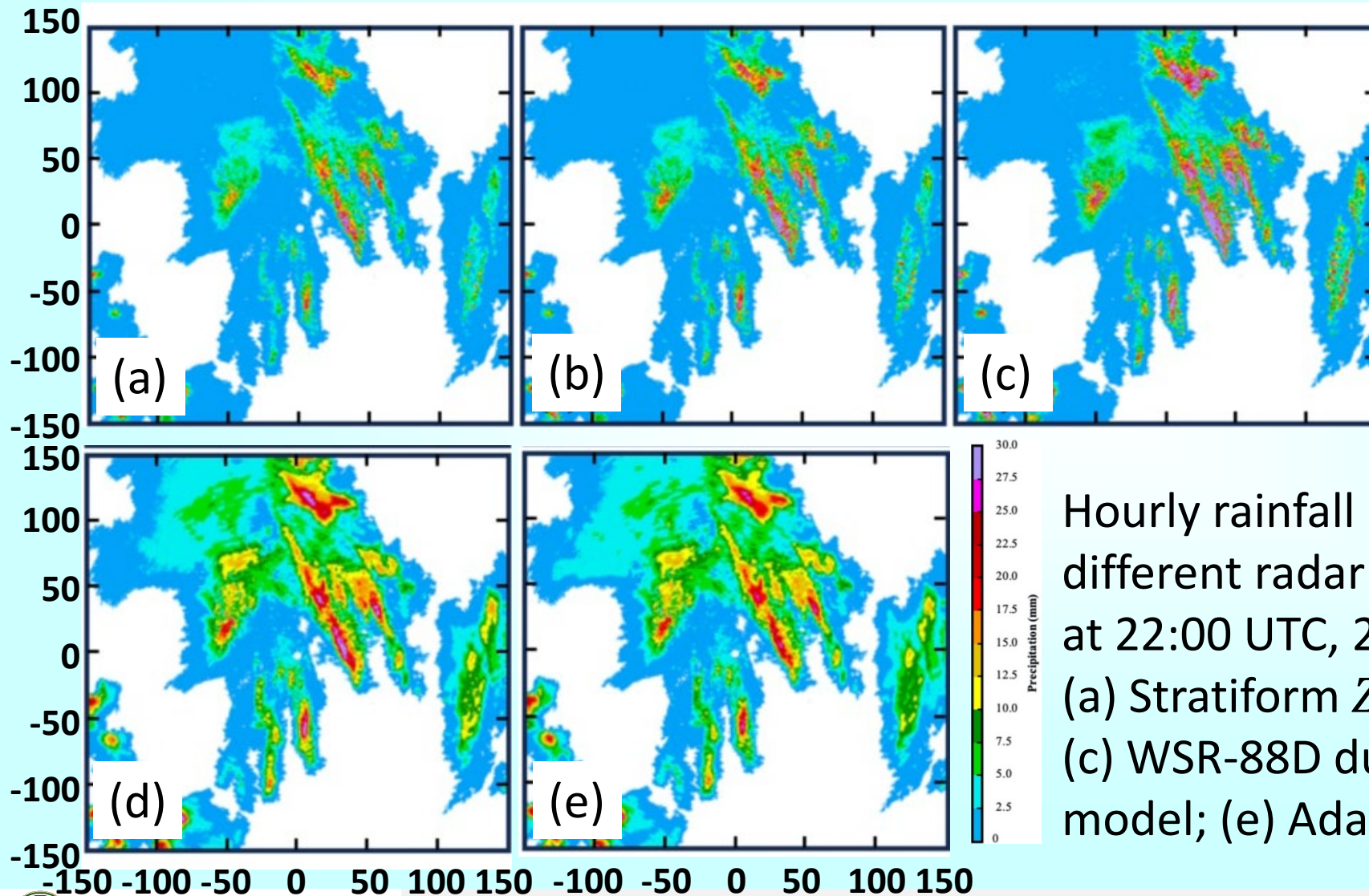
- The long-distance regulation module uses target domain data as input, includes frozen layers  $F_x$ , fine-tuning layers, and an  $\alpha$  adaptive layer.
- During fine-tuning on the target task  $T_T$ , the parameters  $\theta_S$  of the pre-trained model's initial layers (feature extraction part) remain fixed, while only the parameters  $\theta_T^*$  of the final layers (regressor part) are fine-tuned.

where  $L_{ST}(X_T, Y_T)$  represents the estimation loss of the long-distance regulation module.  $L_{MMD}(X_S, Y_S)$  represents the estimation loss of the source domain in the dual-branch model.

**Domain-adaptive loss:**  $LOSS_{DA} = L_{ST}(X_T, Y_T) + L_{MMD}(X_S, Y_S) + \lambda MMD^2 \left( \phi \left( X_I^S \right), \phi \left( X_J^T \right) \right)$

# Transfer Learning to Improve AI QPE Model Generalization

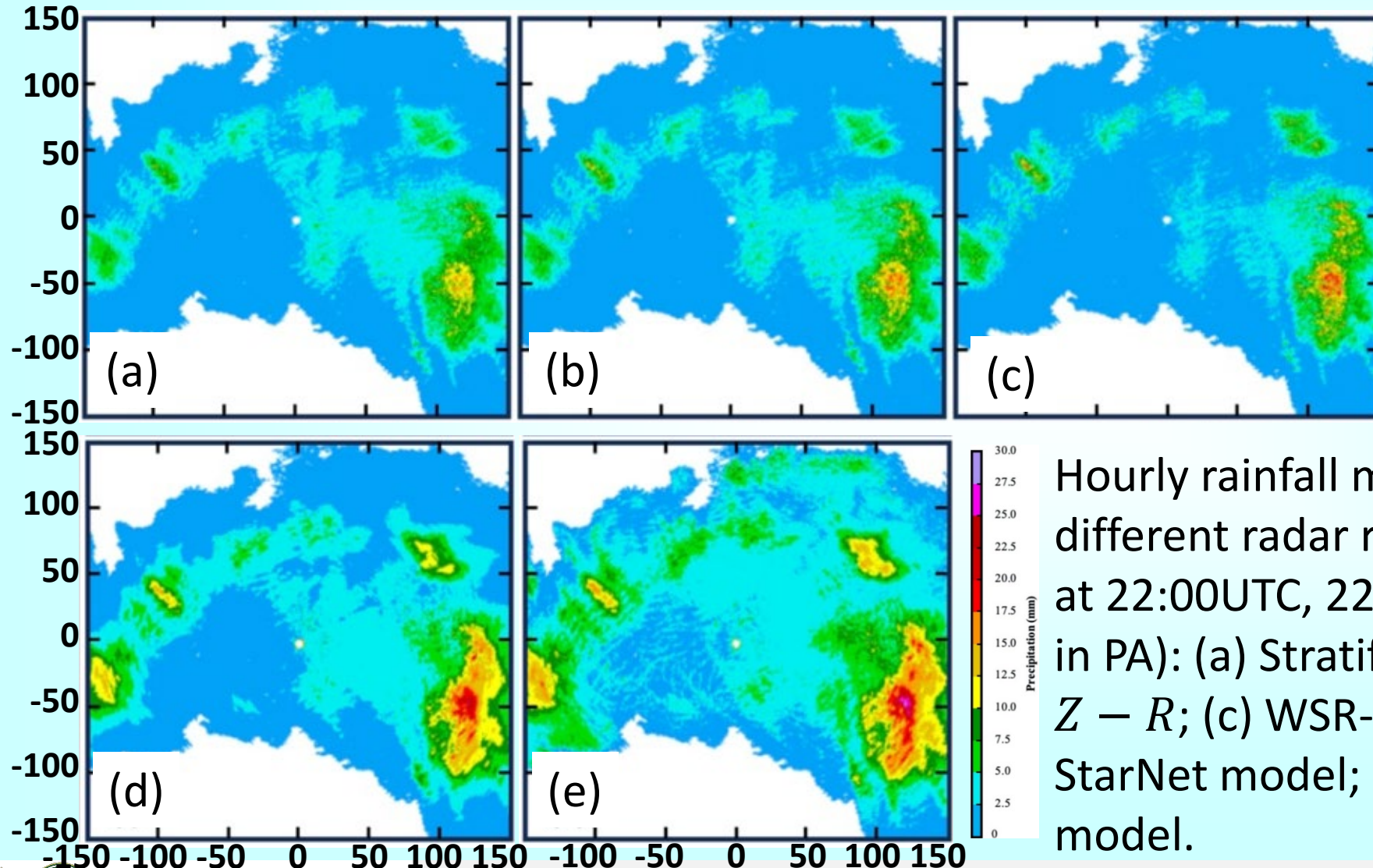
## Demonstration across different precipitation regimes



Hourly rainfall maps generated by different radar rainfall algorithms ending at 22:00 UTC, 29 April 2019 (KESX in NV): (a) Stratiform  $Z - R$ ; (b) Stratiform  $Z - R$ ; (c) WSR-88D dual-pol relation; (d) StarNet model; (e) Adapted StarNet model.

# Transfer Learning to Improve AI QPE Model Generalization

## Demonstration across different precipitation regimes



Hourly rainfall maps generated by different radar rainfall algorithms ending at 22:00UTC, 22 March 2019 (KDIX radar in PA): (a) Stratiform  $Z - R$ ; (b) Stratiform  $Z - R$ ; (c) WSR-88D dual-pol relation; (d) StarNet model; (e) Adapted StarNet model.

# Transfer Learning to Improve AI QPE Model Generalization

## Demonstration across different precipitation regimes

	KDIX Radar*					KESX Radar**				
QPE Algorithm	MAE (mm)	RMSE (mm)	CC	NSE (%)	Bias Ratio	MAE (mm)	RMSE (mm)	CC	NSE (%)	Bias Ratio
Convective $Z - R$	0.99	2.17	0.69	55	0.77	0.67	1.26	0.61	67	1.14
Stratiform $Z - R$	0.88	1.90	0.73	49	0.74	0.65	1.18	0.56	59	0.70
WSR-88D dual-pol relation	0.94	1.91	0.73	53	0.70	0.59	1.08	0.63	54	0.70
<b>StarNet Model</b>	<b>0.60</b>	<b>1.36</b>	<b>0.87</b>	<b>31</b>	<b>0.84</b>	<b>0.42</b>	<b>0.92</b>	<b>0.83</b>	<b>38</b>	<b>0.73</b>
<b>Adapted StarNet Model</b>	<b>0.56</b>	<b>1.29</b>	<b>0.88</b>	<b>31</b>	<b>0.96</b>	<b>0.39</b>	<b>0.85</b>	<b>0.84</b>	<b>36</b>	<b>0.82</b>

\*Performance evaluation based on 5,370 testing samples.

\*\*Performance evaluation based on 443 testing samples.

- AI QPE model performs better than parametric relations.
- Adapted AI model can further improve QPE performance across different domains.

# Summary

- Modern deep learning techniques have been applied in weather radars for about 10 years.
- Successful examples are presented from the perspectives of radar data processing and quantitative precipitation estimation.
- The generalization capability of deep learning can be improved through adapting the model with limited observations in different precipitation regimes.
- Given the rapid development of artificial intelligence and deep learning approaches, applications of weather radar atmospheric remote sensing will be revived.

# Thank you for your attention!

This research was supported by the U.S. National Science Foundation.

Colorado State University  
Haonan.Chen@colostate.edu