Machine Learning Parameterization of a Mature Tropical Cyclone Boundary Layer

Le-Yi Wang and Zhe-Min Tan

School of the Atmospheric Sciences, Nanjing University, Nanjing, China

Background

- Tropical cyclone boundary layer (TCBL) plays an essential role in TC development

(1) **Dissipate TC kinetic energy** & **enhance convergence of water vapor** (CISK; Charney & Eliassen, 1964)

(2) Positive feedback in BL: **sea surface wind ⇆ water vapor flux** (WISHE; Rotunno & Emanuel, 1987; Emanuel 1989, 1995)

- Widely used BL parameterizations are not fit to TCBL

(1) Different BL parameterization results in **vast gap between TC simulations** (Braun & Tao 2000; Nolan et al., 2009; Smith & Thomsen, 2010)

(2) Some unique structures (roll vortexes) in TCBL contribute a lot to turbulent fluxes which are **missed in the current BL parameterizations** (Morrison et al., 2005; Zhang et al., 2008)

Related works

- Cumulus convective parameterization

(1) **Fully connected neural network (FC-NN)** learns from **coarse-grained high resolution simulation** (Krasnopolsky et al., 2013; Brenowitz & Bretherton, 2018, 2019)

(2) **FC-NN** learns from **superparameterization** in GCM (Gentine et al., 2018; Rasp et al., 2018)

(3) **Random forest** learns from **cumulus convective parameterization** and **coarse-grained high resolution simulation** (O'Gorman & Dwyer, 2018; Yuval & O'Gorman, 2020)

- Parameterization in large eddy simulation (LES) and Reynolds-averaged Navier-Stokes equations (RANS)

(1) **FC-NN** learns from **coarse-grained high resolution simulation** (Cheng et al., 2019; Ling et al., 2016; Pawar et al., 2020)

(2) **Random forest** learns to **revise the traditional parameterization** (Wang et al., 2017)

(3) **FC-NN** learns from **computationally costly prarameterization** (Pal 2020)

Basically two paradigm:
(1) Learn from reliable high resolution output **Our choice**
(2) Learn from parameterization and replace them

Large eddy simulation of TCBL

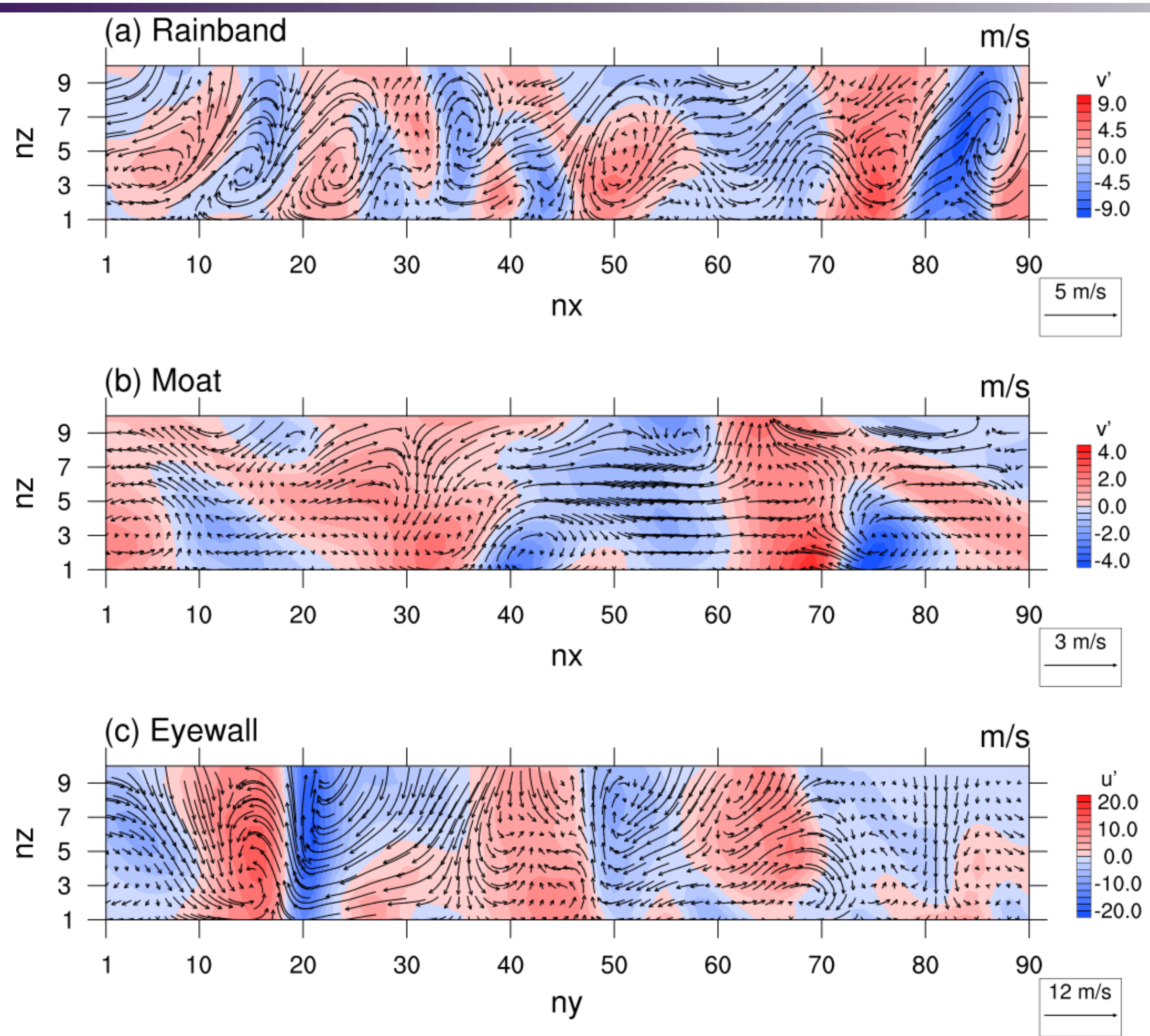


Figure. 1 The vertical cross sections of velocity field in TCBL during TC mature stage. Crossing direction is normal to the long axis of roll vortexes (dx = 100 m, dy = 50 m)

LES Model setup

- WRF v3.9.1.1, idealized simulation with **4 nested domains**
- The inner most domain: **dx = dy = 100 m**, **200 km** side length square
- 69 vertical levels in total with **15 levels below 1200m**
- Whole time span: 119 h, LES grid loaded at 108 h **(11 h for LES)**

Wave length features

(a) 500 m (due to the cluttering effect of convections)

(b) 1-3 km

(c) 1-3 km, intensity much more stronger than in (a) and (b), agree with eye wall observation (Zhang et al., 2011): 0.5 km-3 km

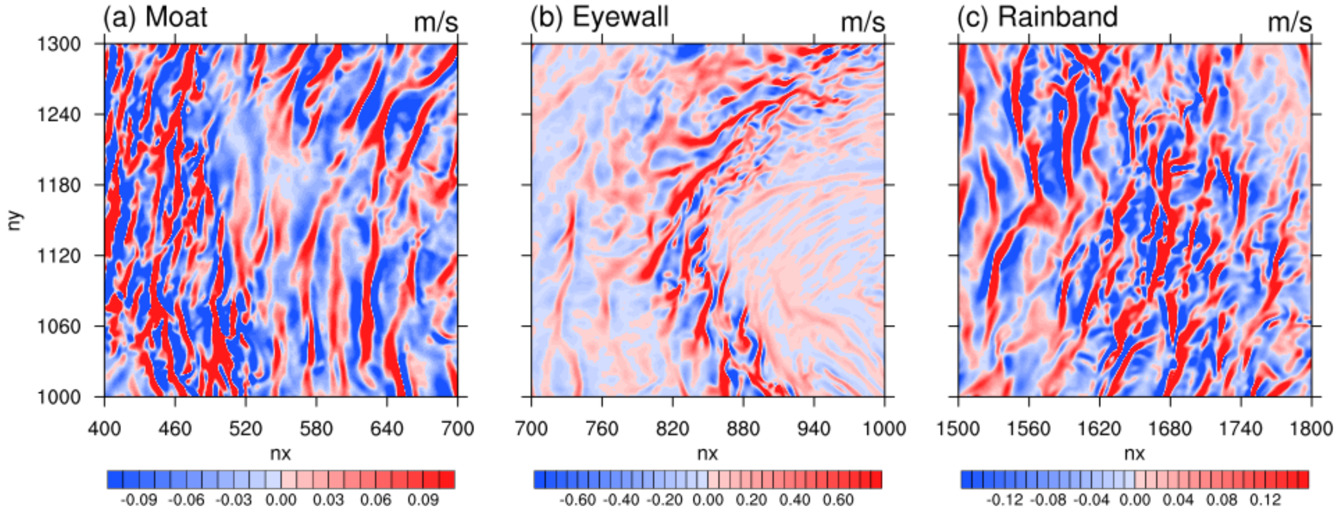


Figure. 2 Vertical wind speed at 27 m in TCBL during TC mature stage

Vertical velocity distribution

(a) Streak-like roll structures featured by velocity pair of upward and downward motions

(b) Stronger than in moat and rainband

(c) More cluttered and disorganized

In agreement with simulation and eyewall observation (Ito et al., 2017; Zhang et al., 2011)

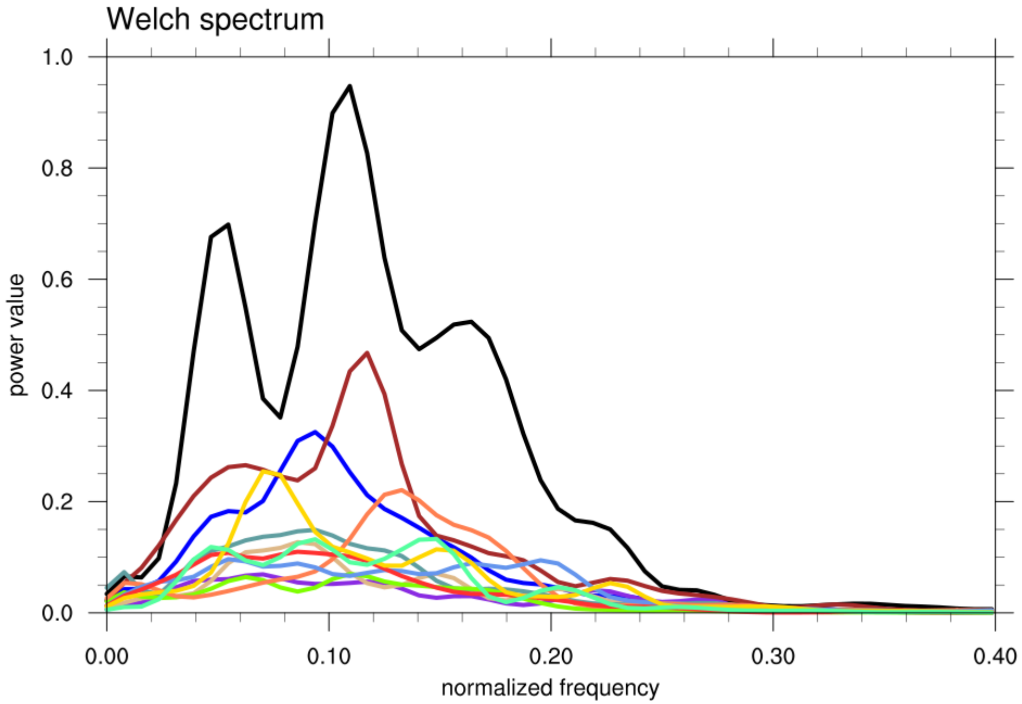


Figure. 3 Welch spectrum analysis of vertical velocity series of roll vortexes along 12 directions in TCBL beyond eyewall (0.05 -> 4 km, 0.2 -> 1 km)

Spectrum analysis

- Roll vortex wave length beyond eyewall is between **1-5 km** which is in perfect agreement with observation (Zhao et al., 2020)
- To sum up, our simulation is **consistent with observations and simulations**, which is convincing enough to develop parameterization

Preprocessing

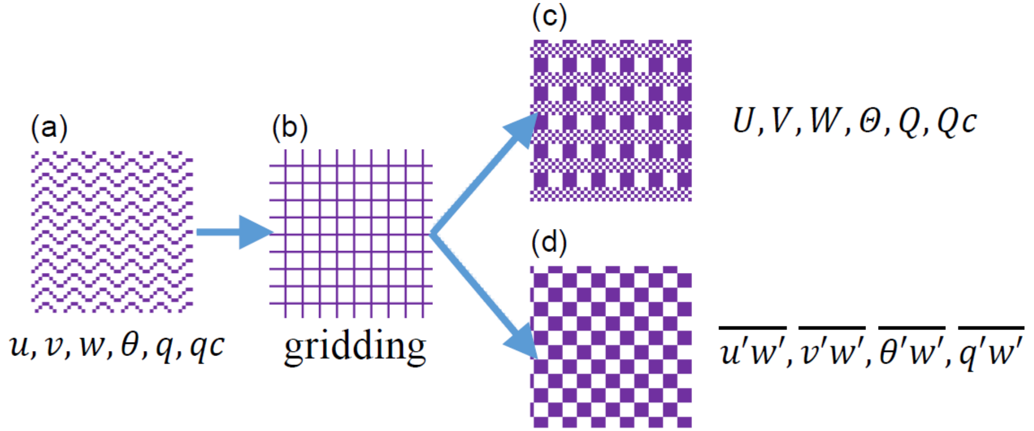


Figure. 4 Flowchart of the derivation of large scale variables and turbulent fluxes

(a) The variables selected: u, v, w, θ, q, qc (q: water vapor mixing ratio; qc: cloud water mixing ratio)

(b) LES domain is divided into many small squared areas **(2 km×2 km)**

(c) Average in each squared area: U, V, W, Θ, Q, Qc

(d) Calculate covariance in each squared area: $\overline{u'w'}$, $\overline{v'w'}$, $\overline{\theta'w'}$ and $\overline{q'w'}$

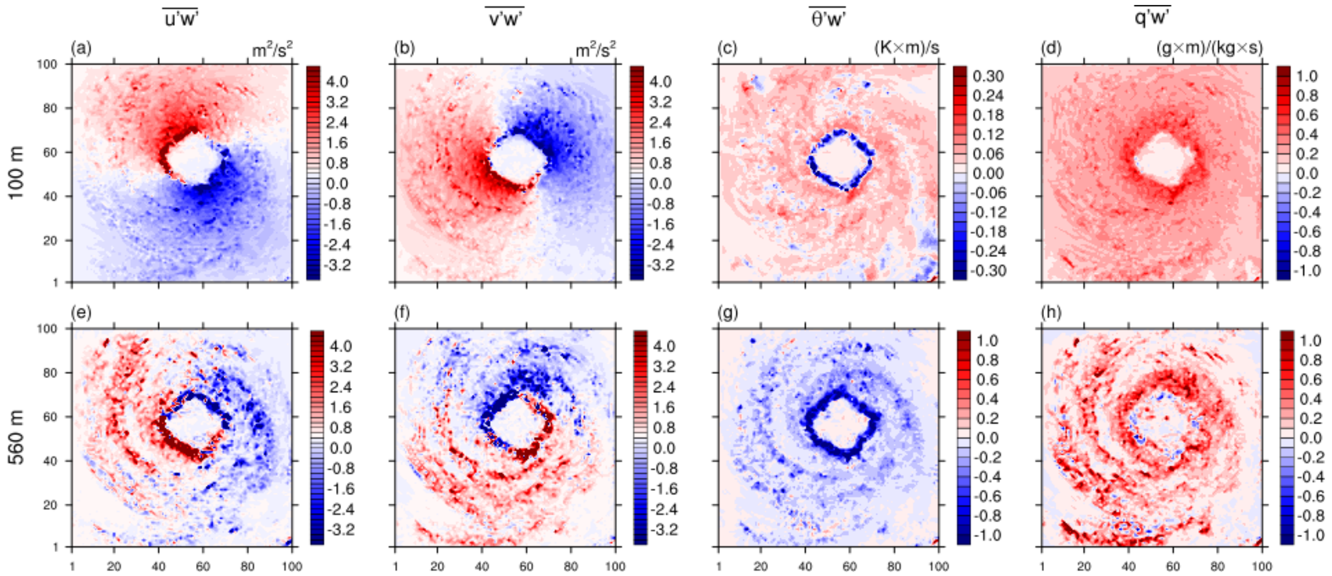


Figure. 5 Turbulent flux of different types at the height of 100 m and 560 m

Flux distribution features

- Low level: eyewall structure, vague rainbands, **symmetric** and relatively **smooth distributions**
- High level: evident eyewall and rainband structures, **convective** and **asymmetric signals**

NN structures and nonlinear transformation

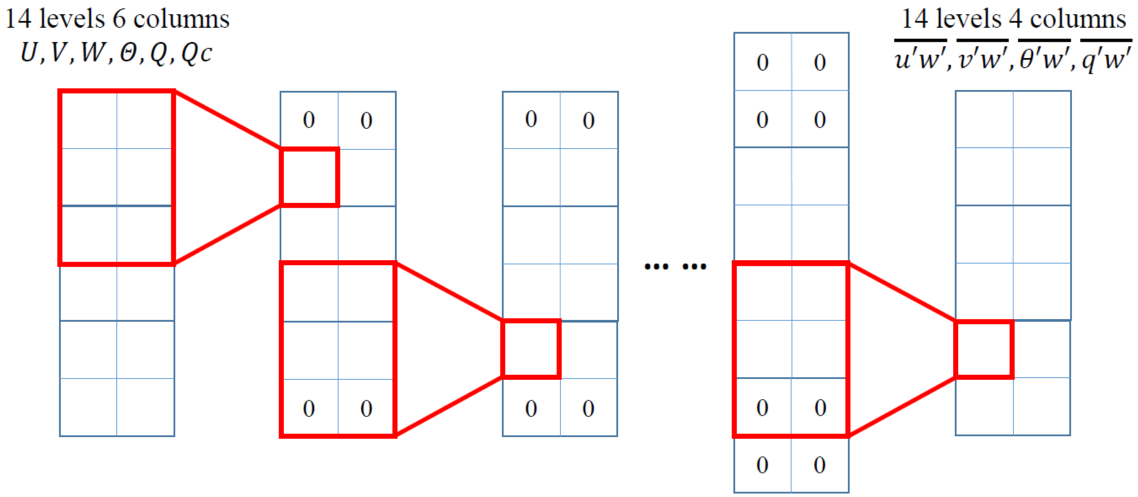


Figure. 6 Schematic diagram of 1D-CNN structure

- NN structure: one-dimension convolutional neural network (1D-CNN)
- All the data are in the same air column in TCBL (the bottom 14 levels)
- Input: 14×6; output: 14×4
- Hidden layer: all have same number of feature maps with zero-padding to conserve NN width
- Kernel size = 3

$$x_{new} = \begin{cases} x^n, & x \geq 0 \\ -(-x)^n, & x < 0 \end{cases} \quad (1)$$

$$\hat{x}_{pred} = \begin{cases} \hat{x}^{1/n}, & \hat{x} \geq 0 \\ -(-\hat{x})^{1/n}, & \hat{x} \leq 0 \end{cases} \quad (2)$$

- Unbalanced distribution will poison the ML model performance (Hensman & Masko, 2015; Pulgar et al., 2017)
- This novel nonlinear transformation can alleviate the unbalanced issue in turbulent fluxes
- Model will be trained on x_{new} [all the turbulent fluxes transformed by (1)]
- Model prediction will be inverse-transformed by (2) to recover the original value

Results

Number of feature maps per layer

Number of layers	36	40	44	48	52	56	60
4	0.645	0.643	0.636	0.635	0.630	0.628	0.627
6	0.552	0.544	0.538	0.537	0.533	0.533	0.527
8	0.521	0.514	0.509	0.507	0.506	0.503	0.503
10	0.509	0.507	0.504	0.502	0.497	0.496	0.496
12	0.512	0.512	0.507	0.508	0.501	0.500	0.498
14	0.517	0.513	0.512	0.511	0.509	0.509	0.507

Table 1. 1D-CNN MSEs on validation set with different super parameter configuration

	$\overline{u'w'}$	$\overline{v'w'}$	$\overline{\theta'w'}$	$\overline{q'w'}$
1D-CNN	0.634	0.608	0.395	0.528
FC-NN-NODE	0.661(4.1%)	0.635(4.2%)	0.413(4.1%)	0.552(4.4%)
FC-NN-COMP	0.665(4.7%)	0.625(2.7%)	0.432(8.4%)	0.544(2.9%)
FC-NN-PARA	0.664(4.5%)	0.629(3.4%)	0.428(7.6%)	0.557(5.2%)

Table 2. The testing set MSE comparison between the best 1D-CNN in Table 1 and other different FC-NNs. Percentages in parentheses are the improvements achieved by 1D-CNN

First 2 h LES data is deprecated; 3-9 h: training; 10 h: validation; 11 h: testing

Mean squared errors (MSEs) on validation set

- The validation set MSE distribution exists **optimal point (10 layers - 56 feature maps)** not like in training set (not shown)
- The **NN depth** has much more influence on 1D-CNN performance than number of feature maps

Comparison with other FC-NN on testing set

FC-NN-NODE: same number of nodes as in 1D-CNN
FC-NN-COMP: same amount of computational cost as 1D-CNN
FC-NN-PARA: same number of free parameters as in 1D-CNN

- On testing set, the 1D-CNN achieves **consistent improvement** compared to traditional FC-NN
- Compared to FC-NN-NODE: **less parameters** leads to easier training
- Compared to FC-NN-COMP and FC-NN-PARA: 1D-CNN achieves **much more nodes and expressivity**

$$score = \frac{\sum_{j=1}^{M}\sum_{k=1}^{H}|turflux(j,k)|}{M \times H} \quad (3)$$

$$M = 4,\ H = 14$$

- *Each air column in the dataset is allocated by a **score***
- *The diagnosed turbulent fluxes are normalized by **stds** and **avgs** calculated **level by level** for different type of fluxes*
- *The bigger the score, the extremer the turbulent fluxes*

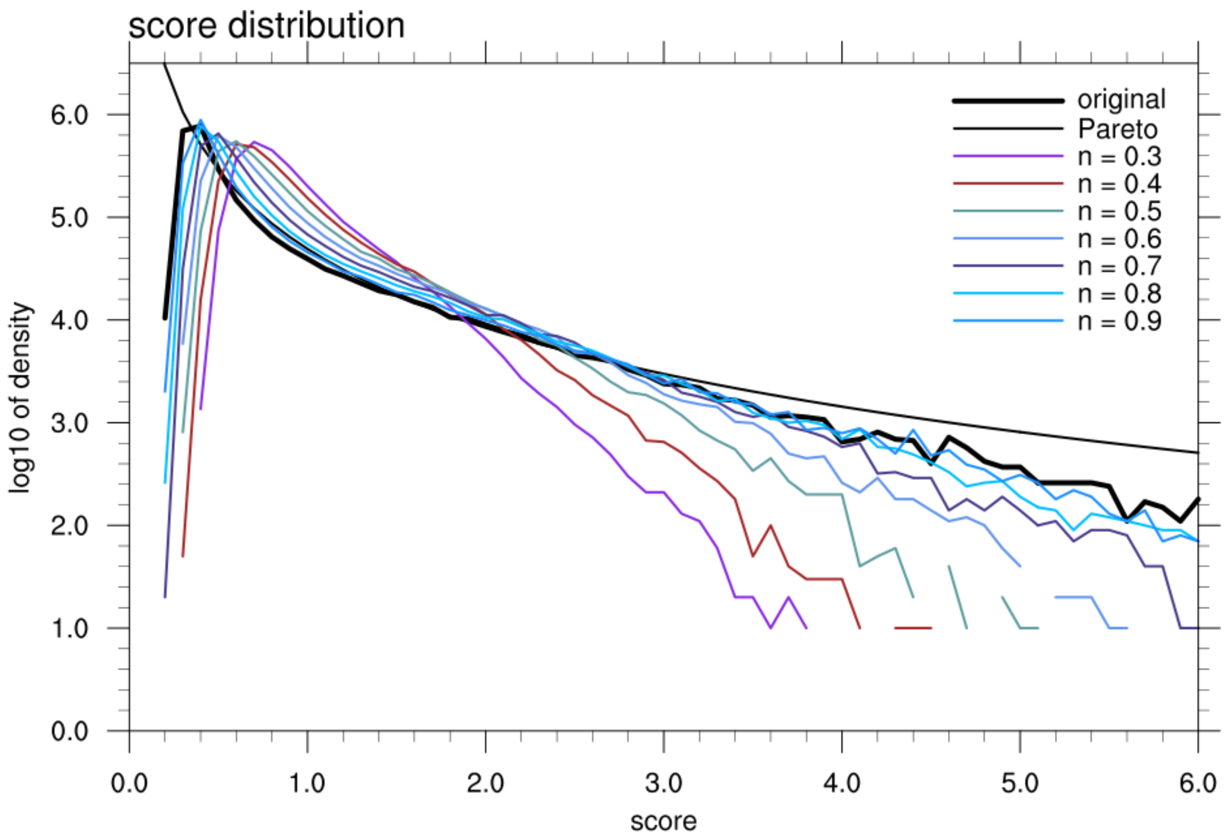


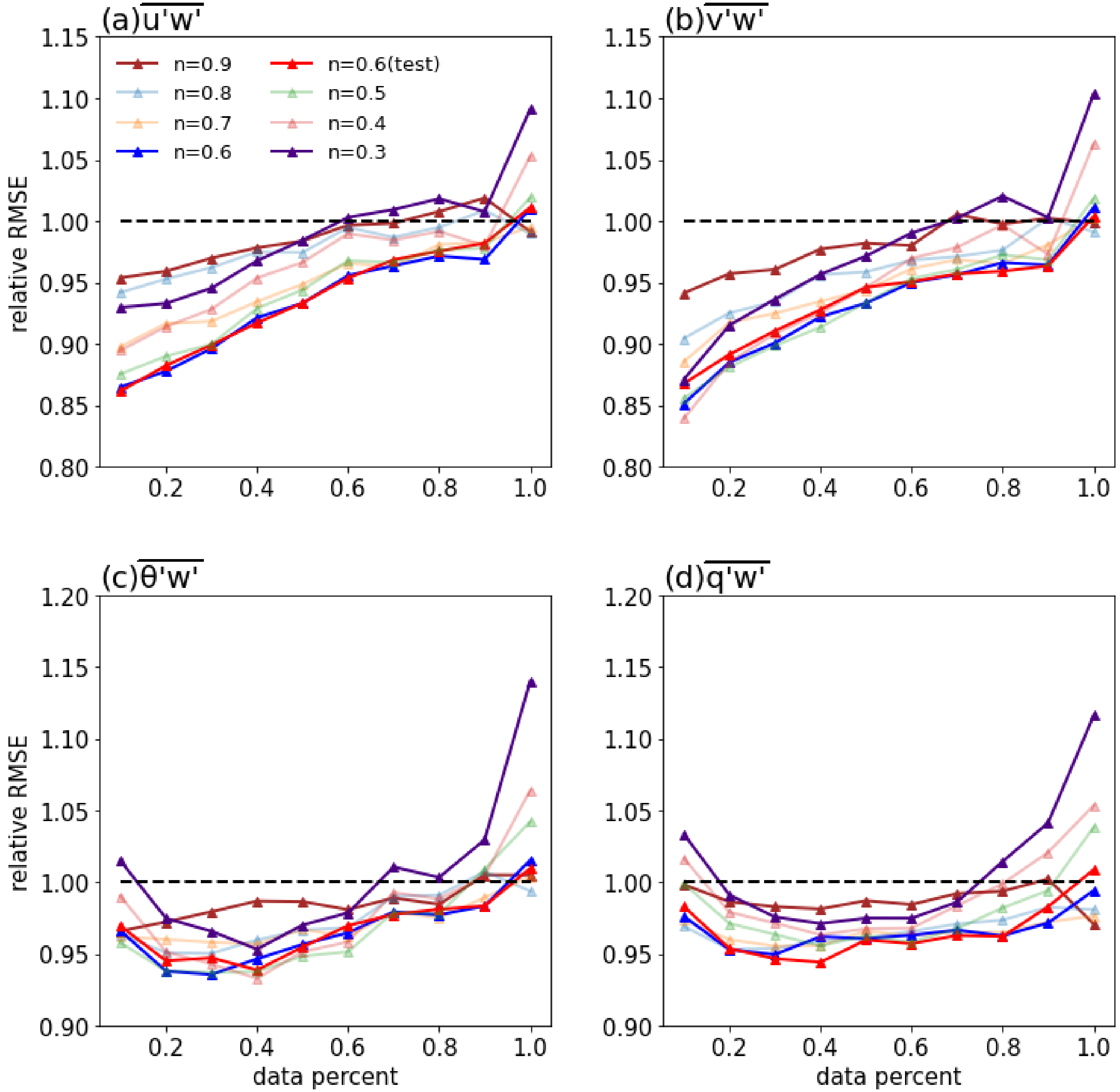
Figure 7. The score distribution of turbulent fluxes on training and validation set before and after the nonlinear transformation (1). n is the power number in (1). The area under the curve is the number of training and validation data.

$$x_{new} = \begin{cases} x^n, & x \geq 0 \\ -(-x)^n, & x < 0 \end{cases} \quad (1)$$

- The original data distribution is very close to Pareto distribution (a widely used long-tail distribution), indicating imbalanced nature of turbulent fluxes
- As n decreases, the distribution is more compact; the extreme value goes down; the score range shrinks

Distribution skewness for different n

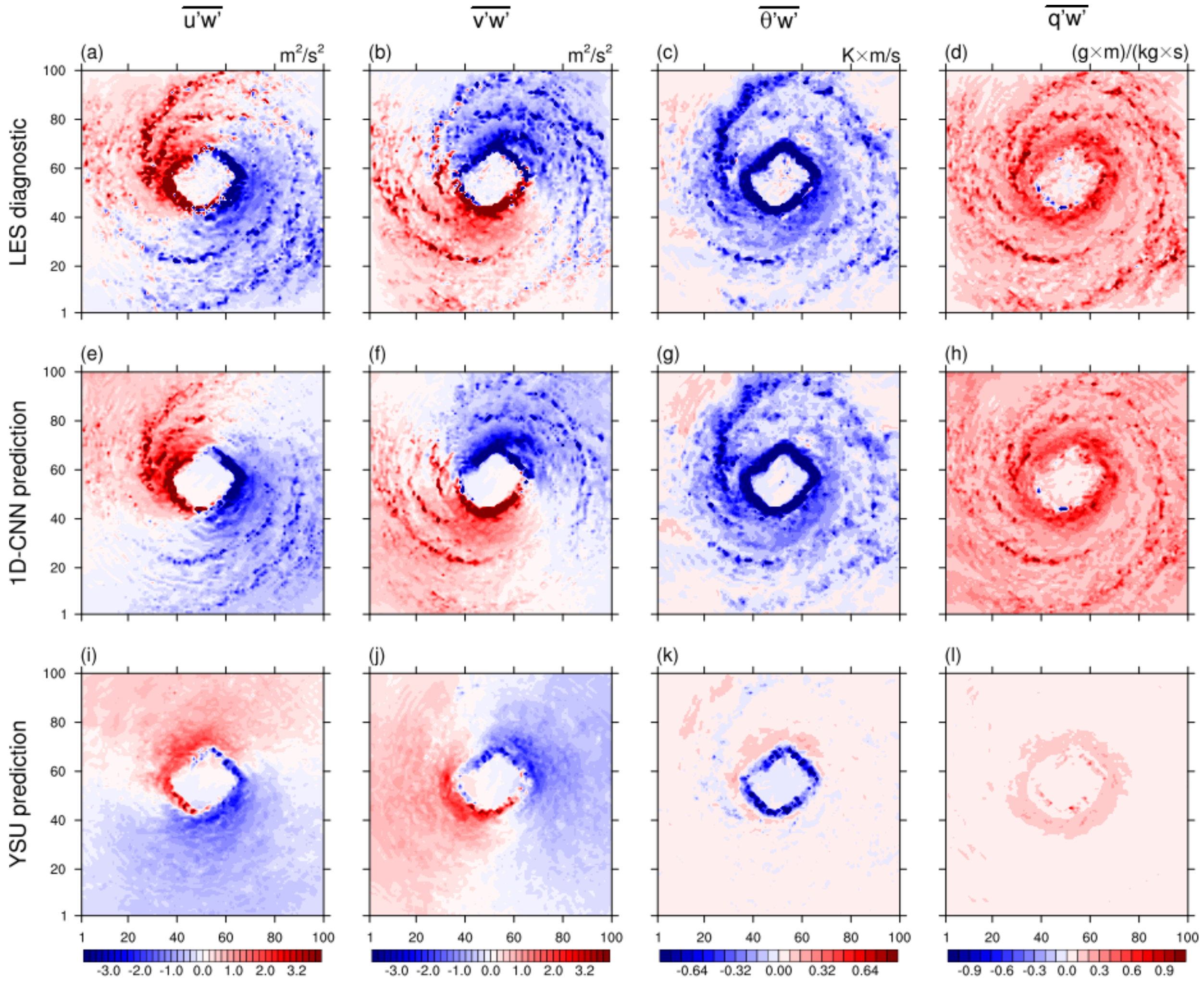
n	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3
skw	4.55	4.08	3.64	3.23	2.85	2.50	2.17	1.90



$$x_{new} = \begin{cases} x^n, & x \geq 0 \\ -(-x)^n, & x < 0 \end{cases} \qquad (1)$$

- *The data on validation or testing set are all divided into ten groups (0.1~1.0). The turbulent flux **extremeness increases from 0.1 to 1.0***
- *The relative root-mean-squared error (RMSE) is the **ratio** between RMSE of 1D-CNN trained on **nonlinearly transformed** data and on **original** data. Each group gets a relative RMSE*
- As n decreases from 1.0 to 0.6, the relative RMSE of 1D-CNN decreases for all the groups, except for the last group
- As n decreases from 0.6 to 0.3, the relative RMSE increases again, and the last group result **explodes**
- The proper n will let 1D-CNN enjoy a **considerable RMSE decrease**, which also applies to the testing set
- **5-15%** for momentum flux, **2-6%** for sensible and water vapor flux

Figure 8. The relative RMSE distribution of 1D-CNN trained on nonlinearly transformed data. The assessment is made on validation set except for n=6 (test), which is on testing set



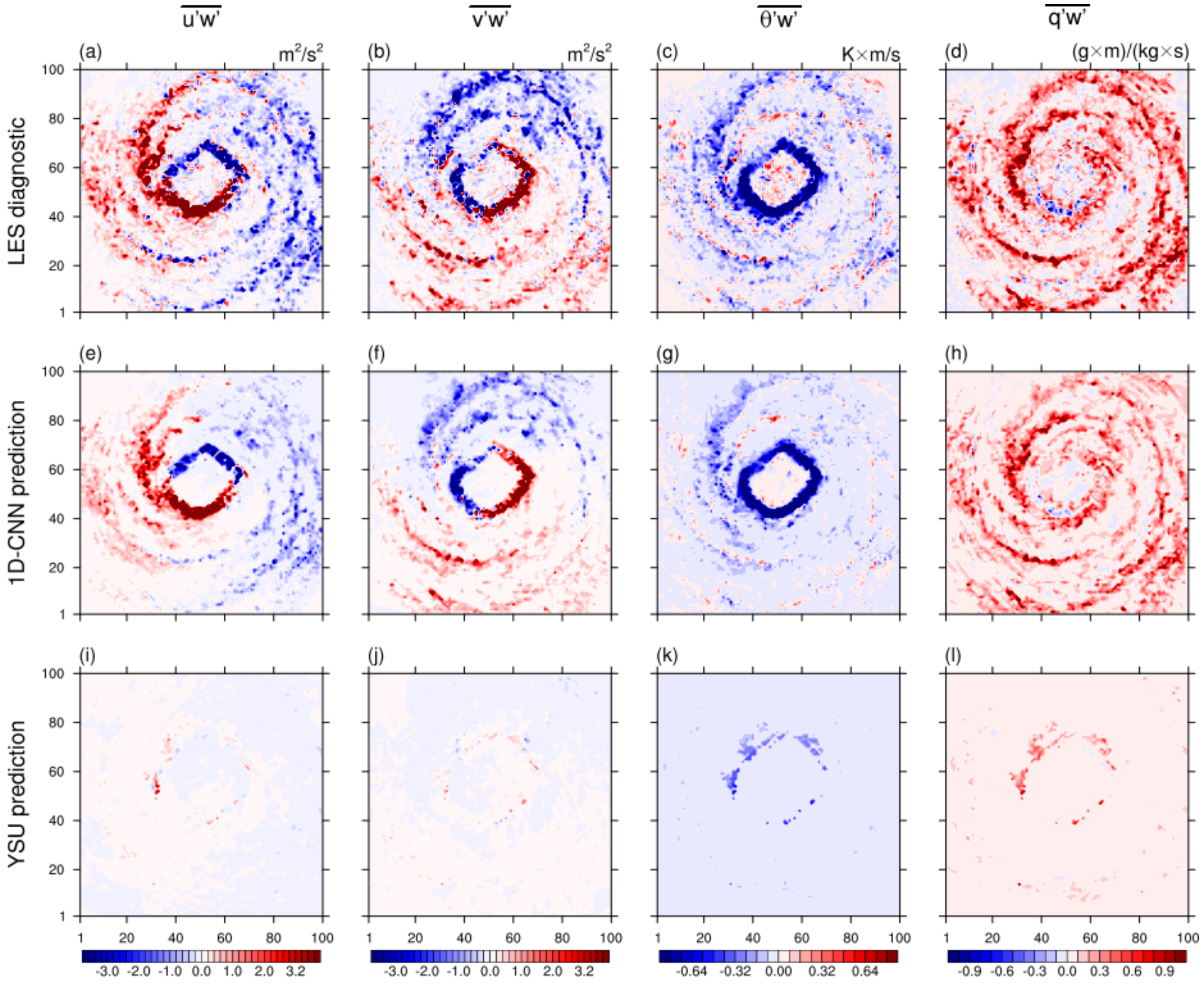
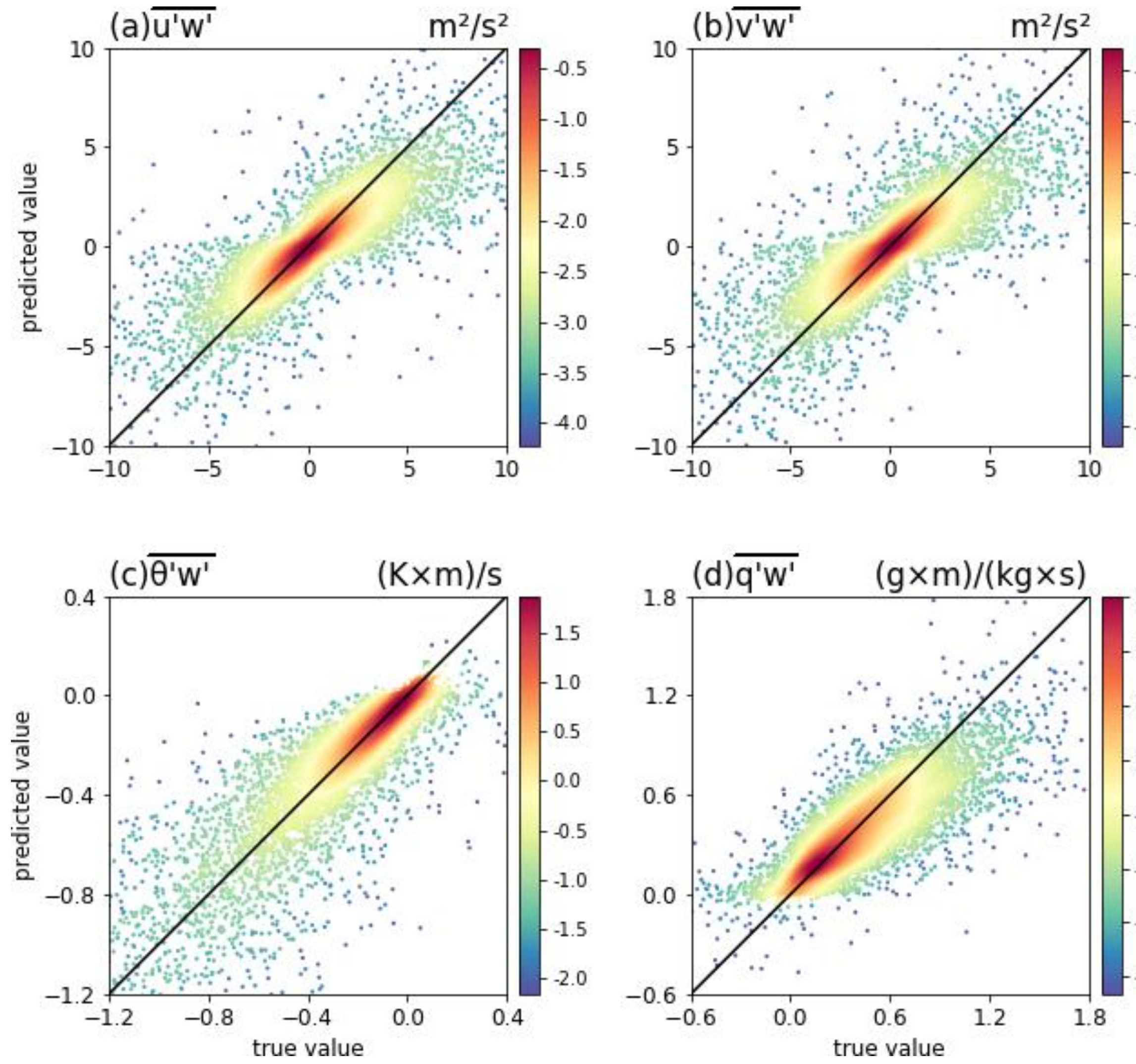


Figure 9 & 10. The spatial distribution of turbulent fluxes predicted by YSU and 1D-CNN at fine-grid time 10 h 40 min at 251 m (left) and 780 m (right), which is in the testing set. The 1D-CNN is trained on the nonlinearly transformed data (n=0.6)

- 1D-CNN can reproduce the **spatial distribution** of turbulent flux including eyewall and rainband structures, while the YSU cannot
- The higher level performance **decreases**, probably due to poor predictability



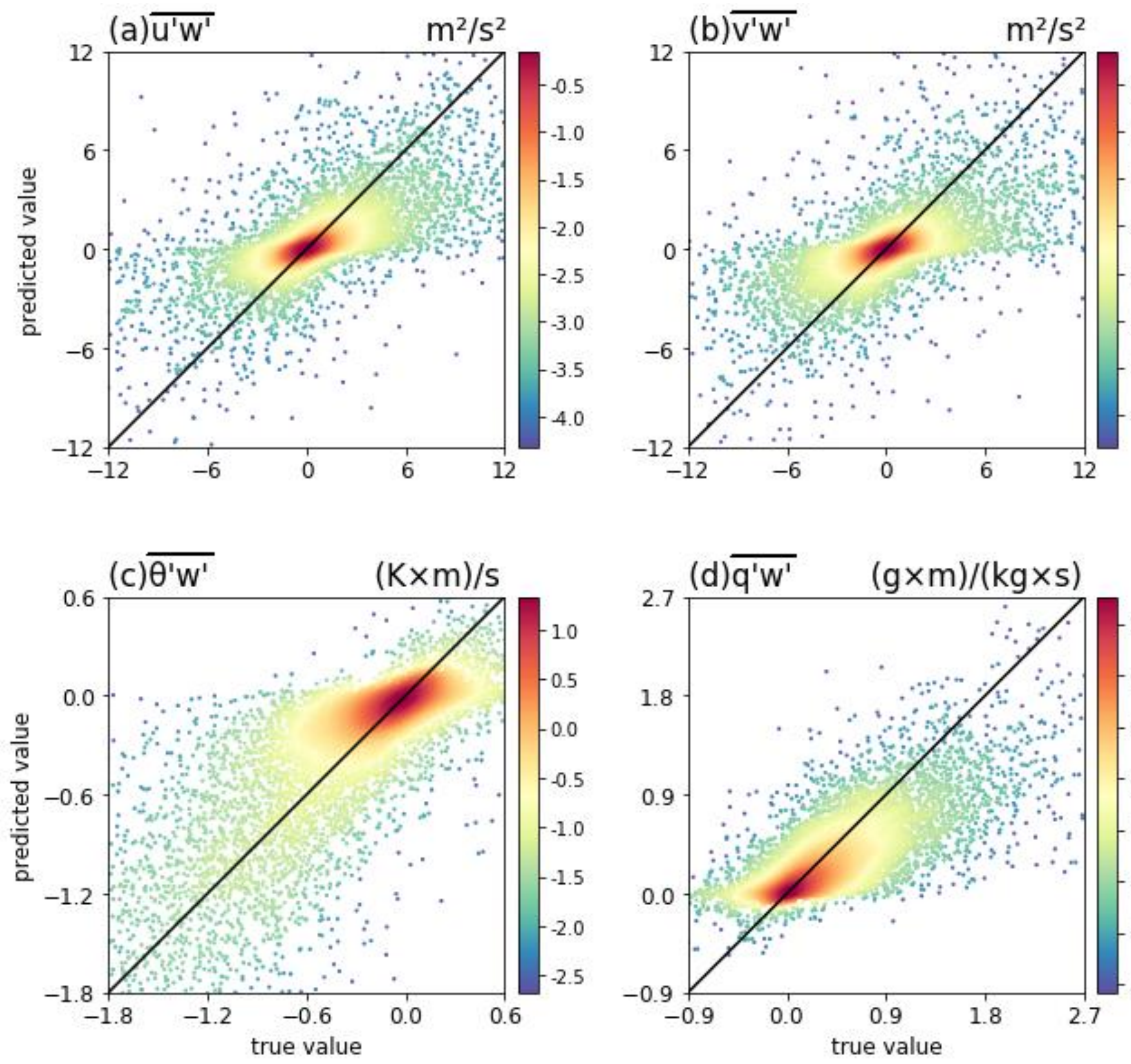


Figure 11 & 12. The prediction scatterplot of the 1D-CNN on the whole testing set at the height of 251 m (left) and 780 m (right). The 1D-CNN is trained on the nonlinearly transformed data (n=0.6)

Same as in Figure 10 and 11, the prediction performance at the higher level is worse. High level suffers from **underestimation**, to be exact. Low level prediction is almost perfect

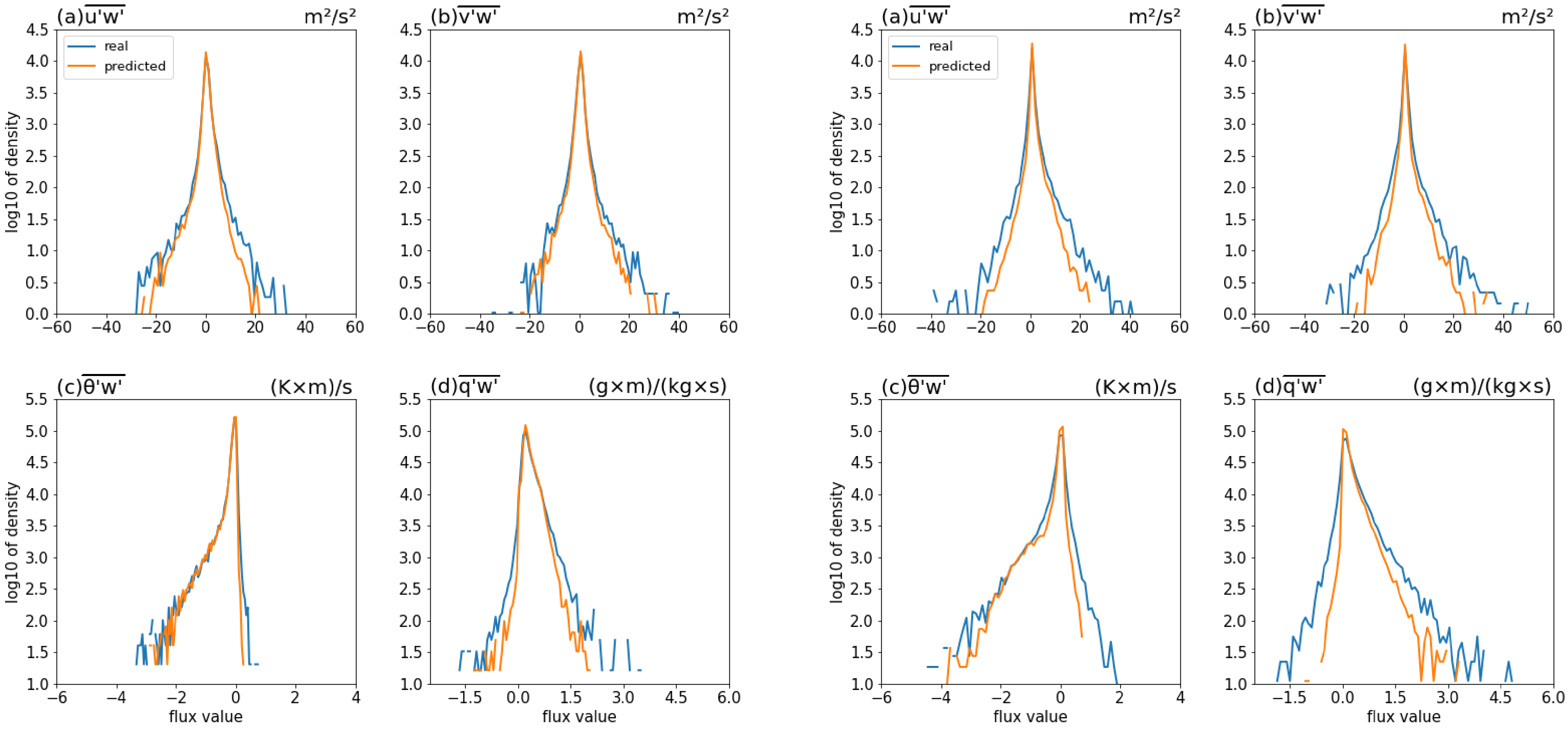


Figure 13 & 14. The original and predicted turbulent flux distribution of the testing set at 251 m (left) and 780 m (right). The 1D-CNN is trained on the nonlinearly transformed data (n=0.6)

1D-CNN achieves almost perfect distribution prediction at low level. A mild **underestimation** of extreme value is observed at high level. The water vapor flux seems hardest to be predicted.

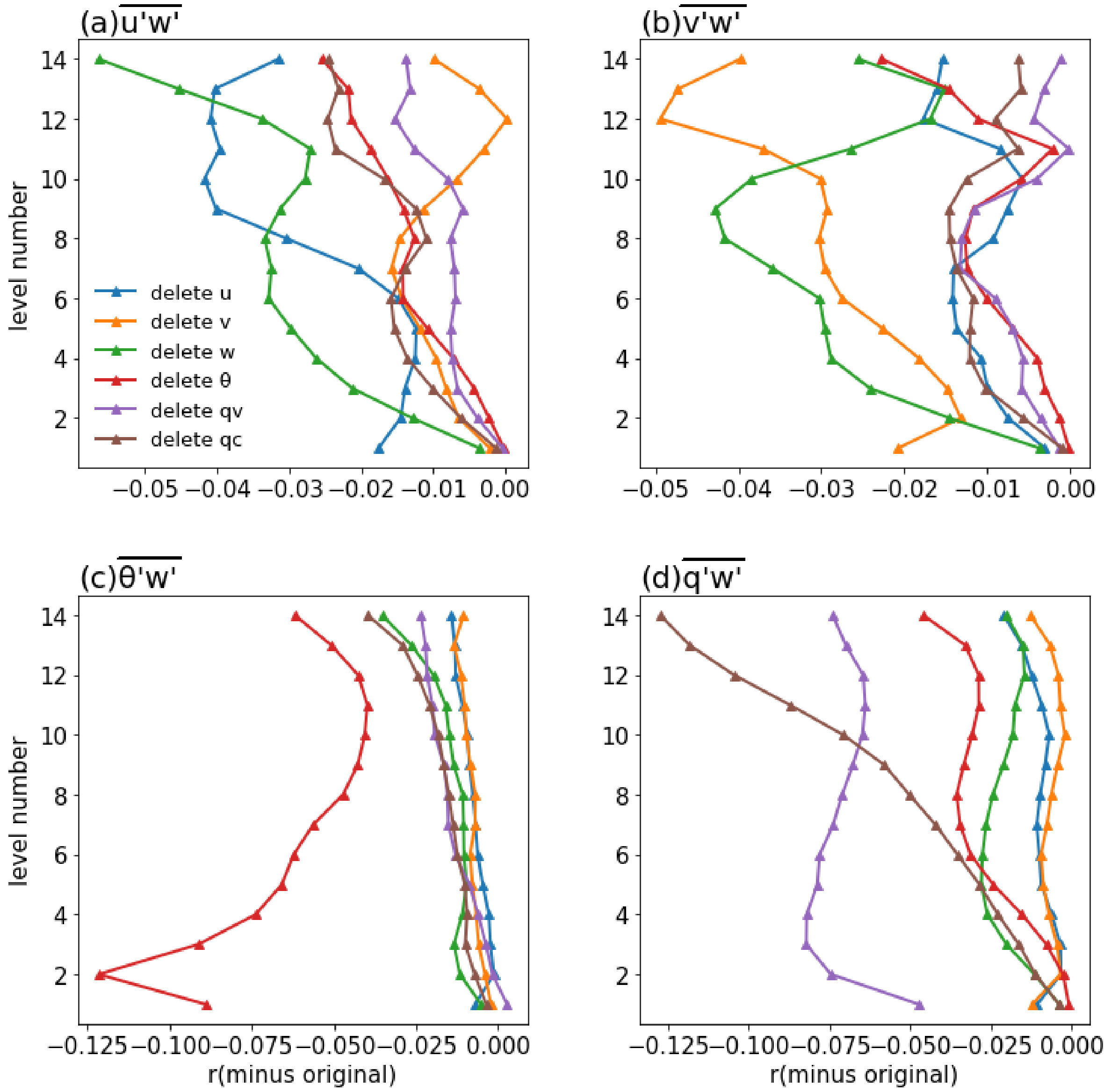


Figure 15. The Pearson correlation coefficient drop when one of the 1D-CNN input variable is deleted

- Most influential input:

- Momentum flux: u and w for $\overline{u'w'}$, v and w for $\overline{v'w'}$

- Sensible heat flux: θ

- Water vapor flux: q and qc

- Unusual high sensibility of u, v, θ and q in low level and w and qc in high level: low level flux in surface layer obeys **Monin-Obukhov similarity** rule, high level flux is more **convective**
- Pearson coefficient drop between level 8 and 12 is not that high: probably due to **stabilizing effects in BL top**

Conclusion and future work

- 1D-CNN achieves **consistent improvements** compared to FC-NN, 1D-CNN has convolution structure which is very convenient for large scale deployment on the GPU in the future
- The novel nonlinear transformation can alleviate the unbalanced nature of turbulent fluxes. This alleviation efficiently **decrease the RMSE** of 1D-CNN
- The trained 1D-CNN successfully reproduces the **spatial and probabilistic distribution** of turbulent flux while the **high level** performance suffers from **underestimation** issue in extreme values
- The **related large-scale variables** have the most influential effect on corresponding turbulent fluxes. Low level flux is more related to **surface variables** while the high level flux is more related to **convections**

Future work

The performance of 1D-CNN on **different resolution**

The performance of 1D-CNN during **online simulation**