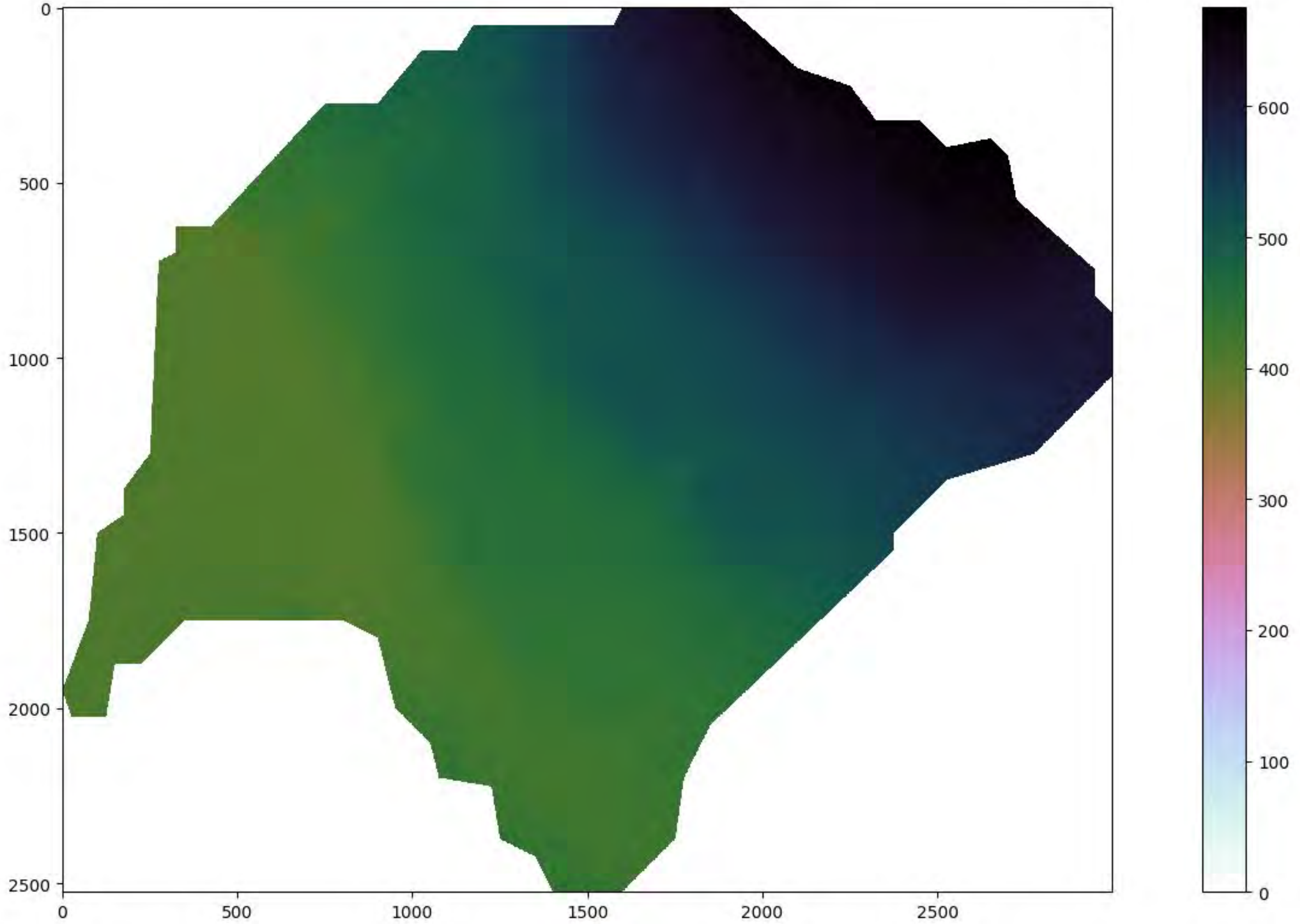
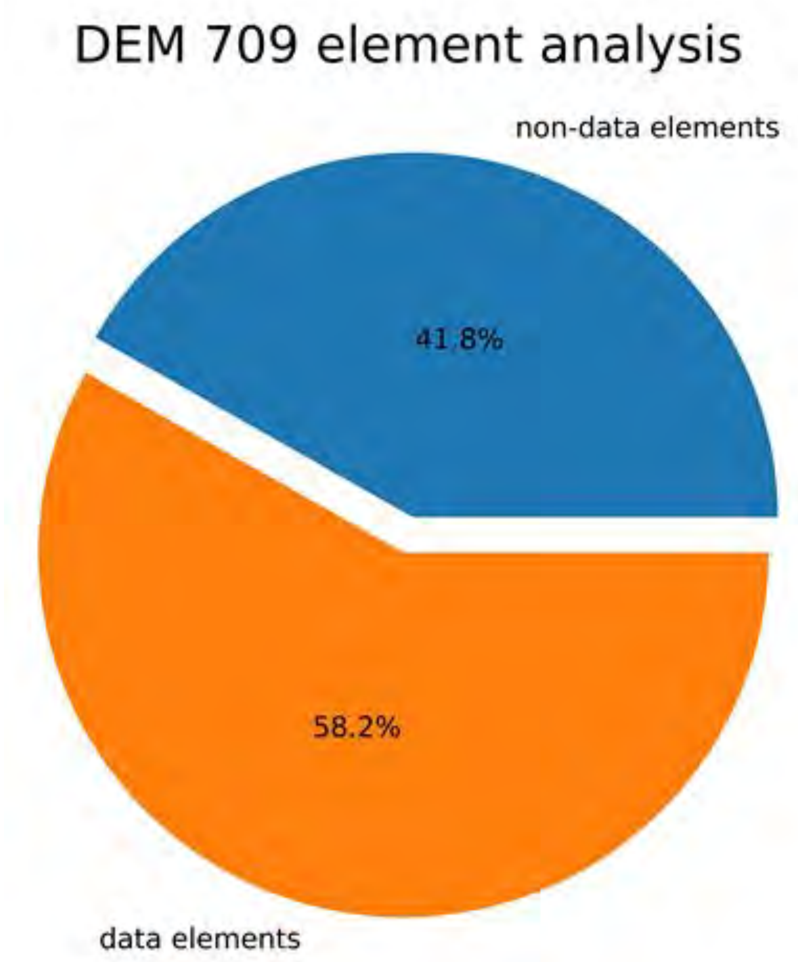
Water level prediction using a multi-task ranking approach

SDSC-EcoVision kick-off "4Real"

Catchment: 709 (2500x3000)





Total elements	7575000
Non-data elements	3165650
Data elements	4409350

Dataset details

Training:

- tr5
- tr20
- tr50
- tr2-2
- tr10-2
- tr20-2
- tr50-2
- tr5-3
- tr10-3
- tr100-3

Validation

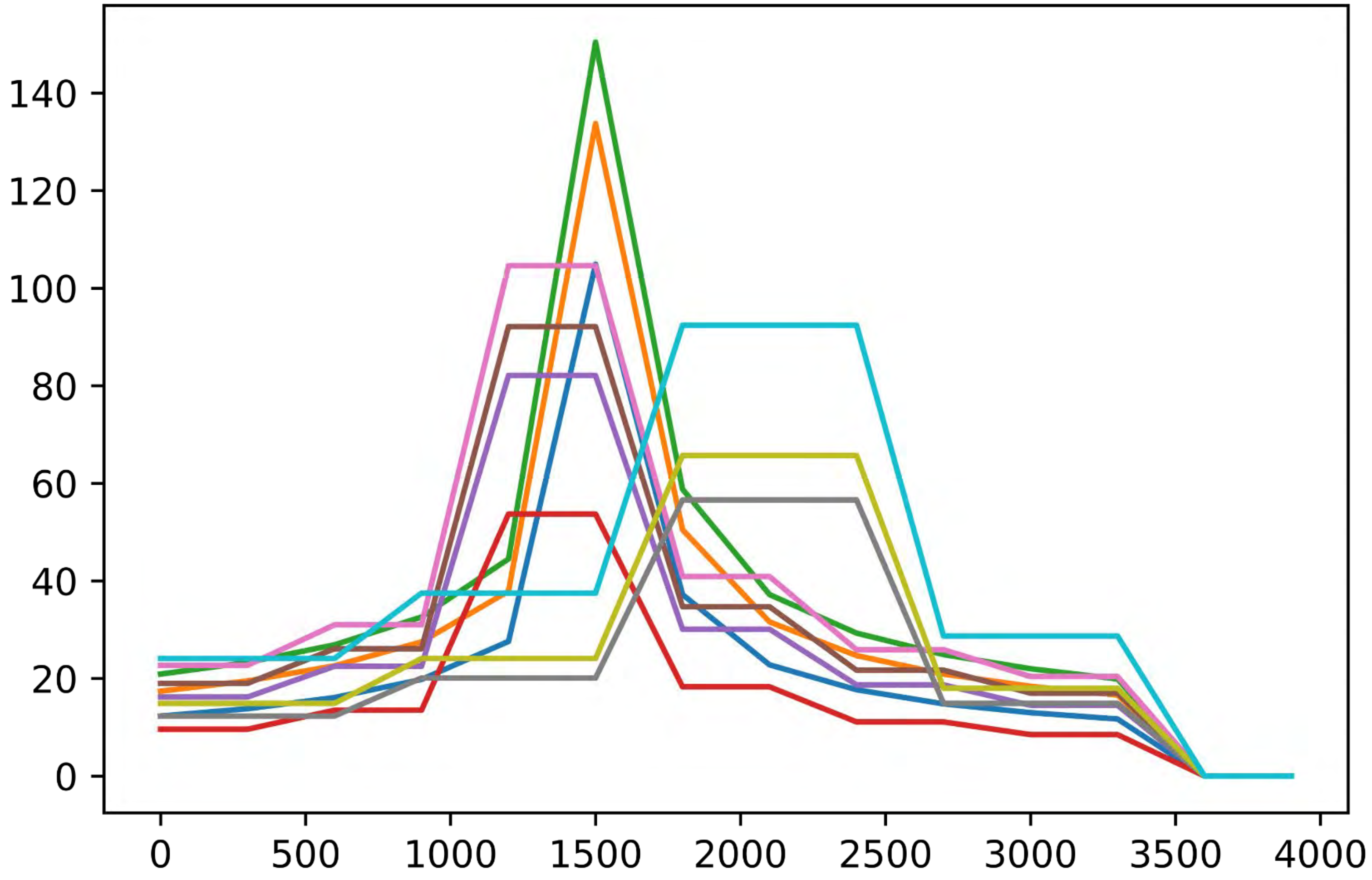
- tr100-2
- tr2-3

Test

- tr2
- tr10
- tr100
- tr5-2
- tr20-3
- tr50-3

Table 1. Hyetographs used for simulations

Name	Test set	Return period	Rainfall intensity (mm/h)											
			0-5 min	5-10 min	10-15 min	15-20 min	20-25 min	25-30 min	30-35 min	35-40 min	40-45 min	45-50 min	50-55 min	55-60 min
tr2	**yes**	2	8.7	9.9	11.5	14.3	20.1	80.1	27.3	16.5	12.7	10.6	9.2	8.3
tr5	no	5	12.3	13.8	16.1	19.8	27.6	104.9	37.2	22.8	17.7	14.8	13.0	11.7
tr10	**yes**	10	14.9	16.7	19.4	23.8	33.0	120.1	44.1	27.3	21.3	17.9	15.7	14.2
tr20	no	20	17.4	19.5	22.6	27.5	37.9	133.7	50.5	31.6	24.7	20.9	18.4	16.6
tr50	no	50	20.9	23.3	26.9	32.6	44.5	150.4	58.8	37.2	29.3	24.9	22.0	19.9
tr100	**yes**	100	24.1	26.8	30.7	37.0	50.1	161.4	65.6	42.1	33.4	28.6	25.3	23.0
tr2-2	no	2	9.6	9.6	13.5	13.5	53.7	53.7	18.3	18.3	11.1	11.1	8.5	8.5
tr5-2	**yes**	5	13.4	13.4	18.7	18.7	71.1	71.1	25.2	25.2	15.4	15.4	12.0	12.0
tr10-2	no	10	16.2	16.2	22.5	22.5	82.1	82.1	30.1	30.1	18.7	18.7	14.5	14.5
tr20-2	no	20	19.0	19.0	26.1	26.1	92.1	92.1	34.7	34.7	21.7	21.7	17.0	17.0
tr50-2	no	50	22.7	22.7	31.0	31.0	104.6	104.6	40.9	40.9	25.9	25.9	20.4	20.4
tr100-2	no	100	26.1	26.1	35.2	35.2	113.5	113.5	46.1	46.1	29.6	29.6	23.5	23.5
tr2-3	no	2	8.8	8.8	8.8	14.5	14.5	14.5	42.5	42.5	42.5	10.7	10.7	10.7
tr5-3	no	5	12.3	12.3	12.3	20.1	20.1	20.1	56.6	56.6	56.6	14.9	14.9	14.9
tr10-3	no	10	14.9	14.9	14.9	24.1	24.1	24.1	65.7	65.7	65.7	18.0	18.0	18.0
tr20-3	**yes**	20	17.5	17.5	17.5	27.9	27.9	27.9	74.0	74.0	74.0	21.0	21.0	21.0
tr50-3	**yes**	50	20.9	20.9	20.9	33.1	33.1	33.1	84.6	84.6	84.6	25.0	25.0	25.0
tr100-3	no	100	24.1	24.1	24.1	37.5	37.5	37.5	92.4	92.4	92.4	28.7	28.7	28.7


140
120
100
80
60
40
20
0
0
500
1000
1500
2000
2500
3000
3500
4000

Experiment details

- **Architectures: UNet, ResNet-34**
- Masking of non-data cell values
- Input: 3 channels, (256 x 256 x 3)
- Objective function: L1 loss

DEM	$water\_depth_{(t1)}$	dem $rainfall_{(t1)}$

- Output:
  - 1 channel (256 x 256)
  - $water\_depth_{(t2)}$

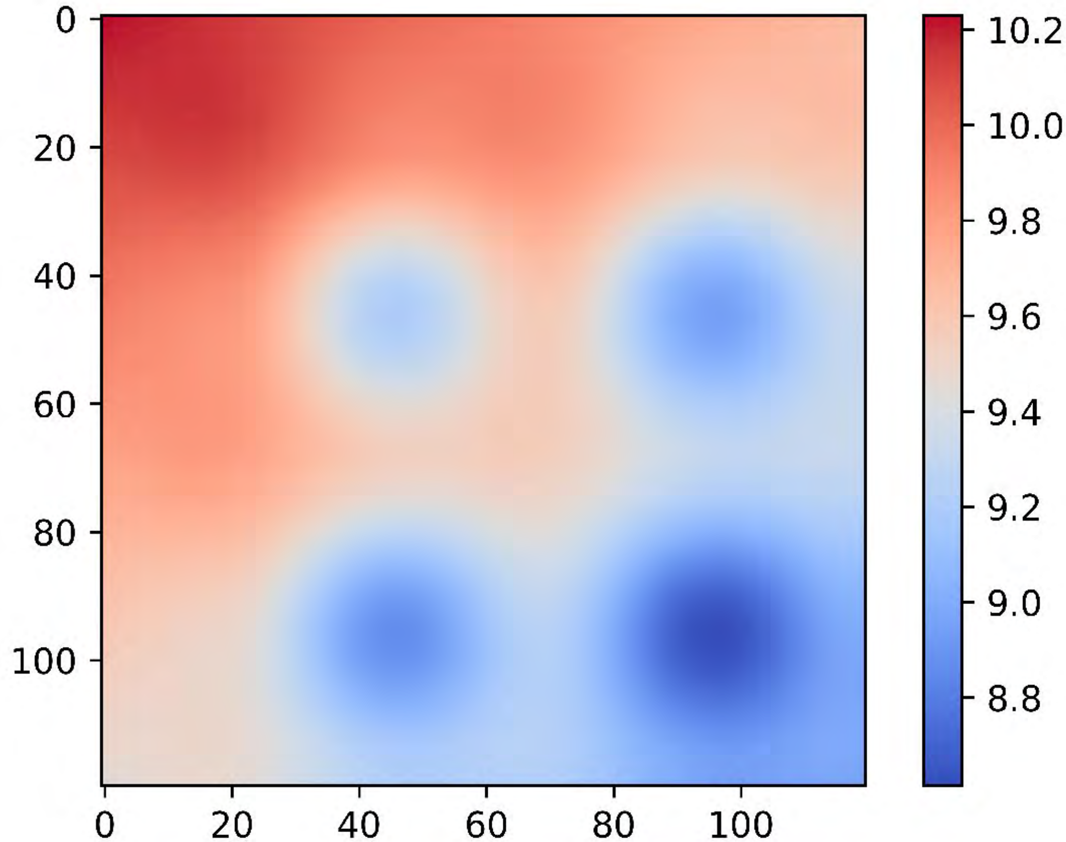
Experiments

- With/without x, y coordinates
- With/without gradient

- **Baseline**:
  - One timestep baseline
    - with input taken as the prediction
  - Two-time step baseline
    - pred = WD_2 + (WD_2 - WD_1)
    - WD: water depth

- One timestep with gradient features

Toy Catchment



Same experiments repeated here

Performance Evaluation

- *p* ← prediction array, *q* ← ground truth array
- *t* ← timesteps, *sum_error* ← sum of errors

sum_error = 0

for all *t*:

  load *p*

  load *q*

  a ← take absolute difference of *p* and *q*

  take sum of all values in *a*

  mean_step ← sum/ number of elements

  add *mean_step* to *sum_error*

One timestep experiments - Toy catchment

Experiment	tr2 (MAE m)	tr10 (MAE m)	tr100 (MAE m)	tr5_2 (MAE m)	tr20_3 (MAE m)	tr50_3 (MAE m)	Average (MAE m)
NN with x, y coordinates	0.0163	0.0176	0.0230	0.0179	0.0192	0.0209	0.0191
NN without x,y coordinates	0.0122	0.0145	0.0215	0.0146	0.0189	0.0211	**0.0171**
Baseline 1timestep	0.0332	0.0535	0.0726	0.0451	0.0593	0.0668	0.0551
NN with gradient	0.0213	0.0305	0.0491	0.0283	0.0350	0.0402	0.0341

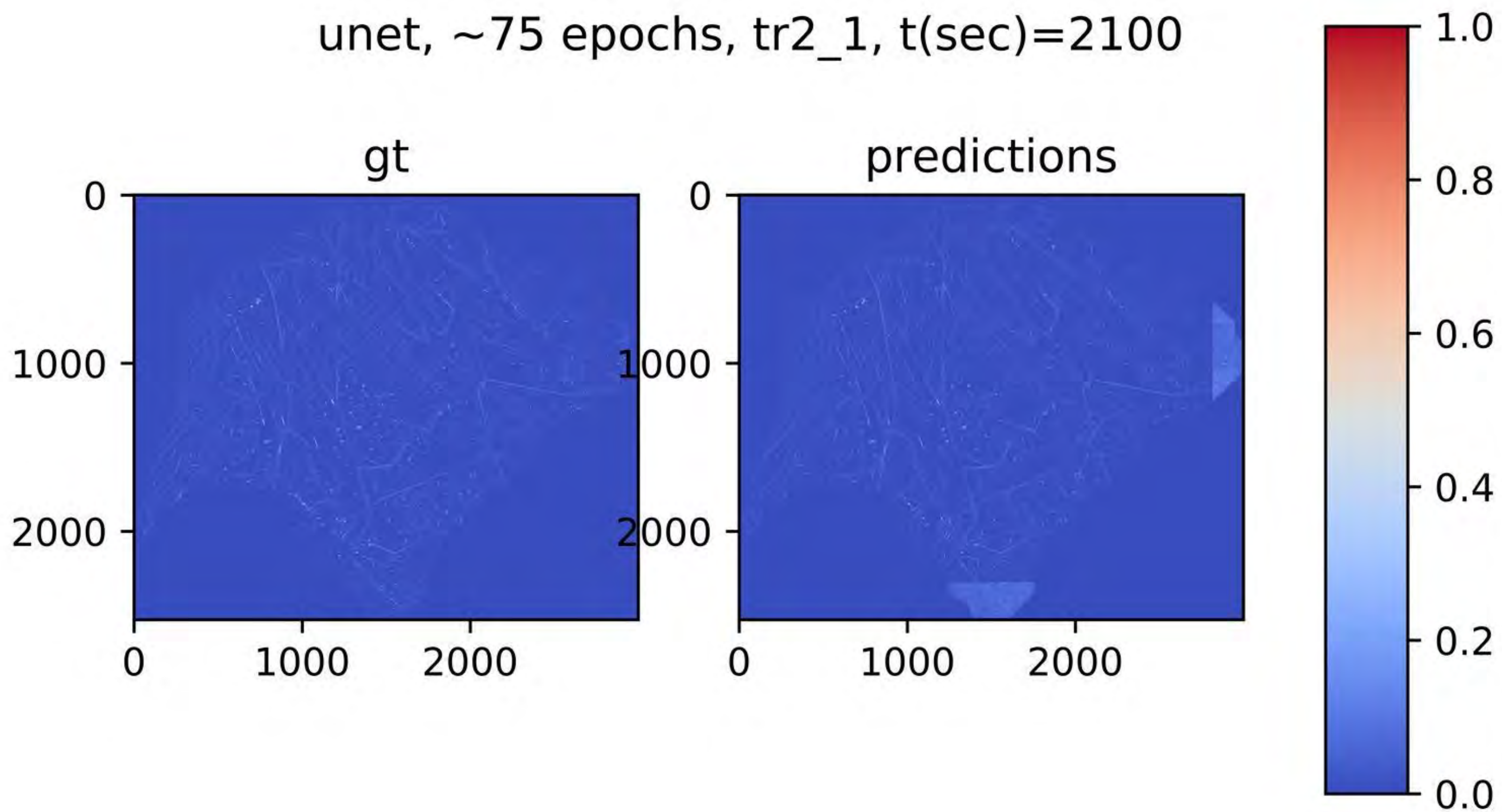
One timestep experiments – 709 (new code)

Experiment	tr2 (MAE m)	tr10 (MAE m)	tr100 (MAE m)	tr5_2 (MAE m)	tr20_3 (MAE m)	tr50_3 (MAE m)	Average (MAE m)
unet, ~75, without_xy	0.0611	0.0766	0.0904	0.0734	0.0760	0.0837	0.0769
Baseline 1timestep	0.0358	0.0593	0.0850	0.0486	0.0653	0.0755	**0.0616**
NN without x,y coordinates	0.1164	0.1417	0.1534	0.1376	0.1383	0.1497	0.1395
UResNet, ~80, ts=1, without_xy	0.0683	0.0895	0.1052	0.0848	0.0855	0.0961	0.0882

One timestep experiments – 709 (new code)

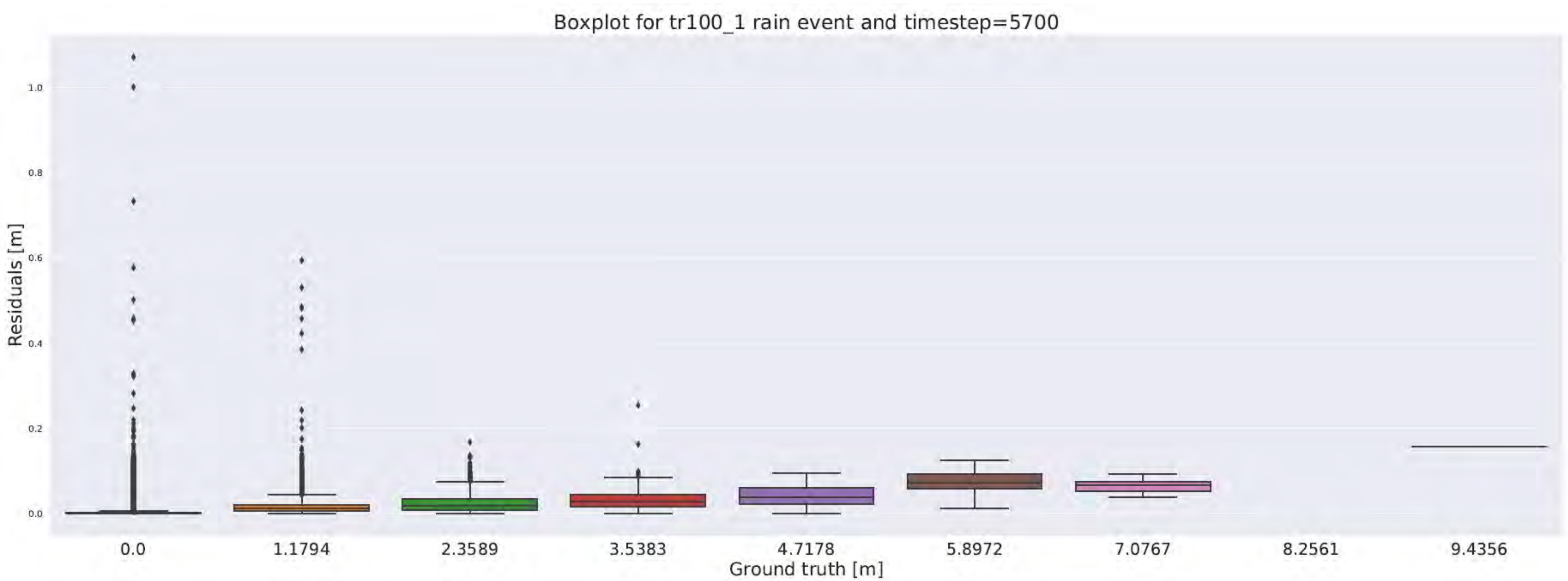
- We see improvement on performance with new updated code on 709 catchment
- The performance of Unet is better than Uresnet
- The baseline performance where we take input as our output for a timestep still outperforms our models.

EXAMPLE

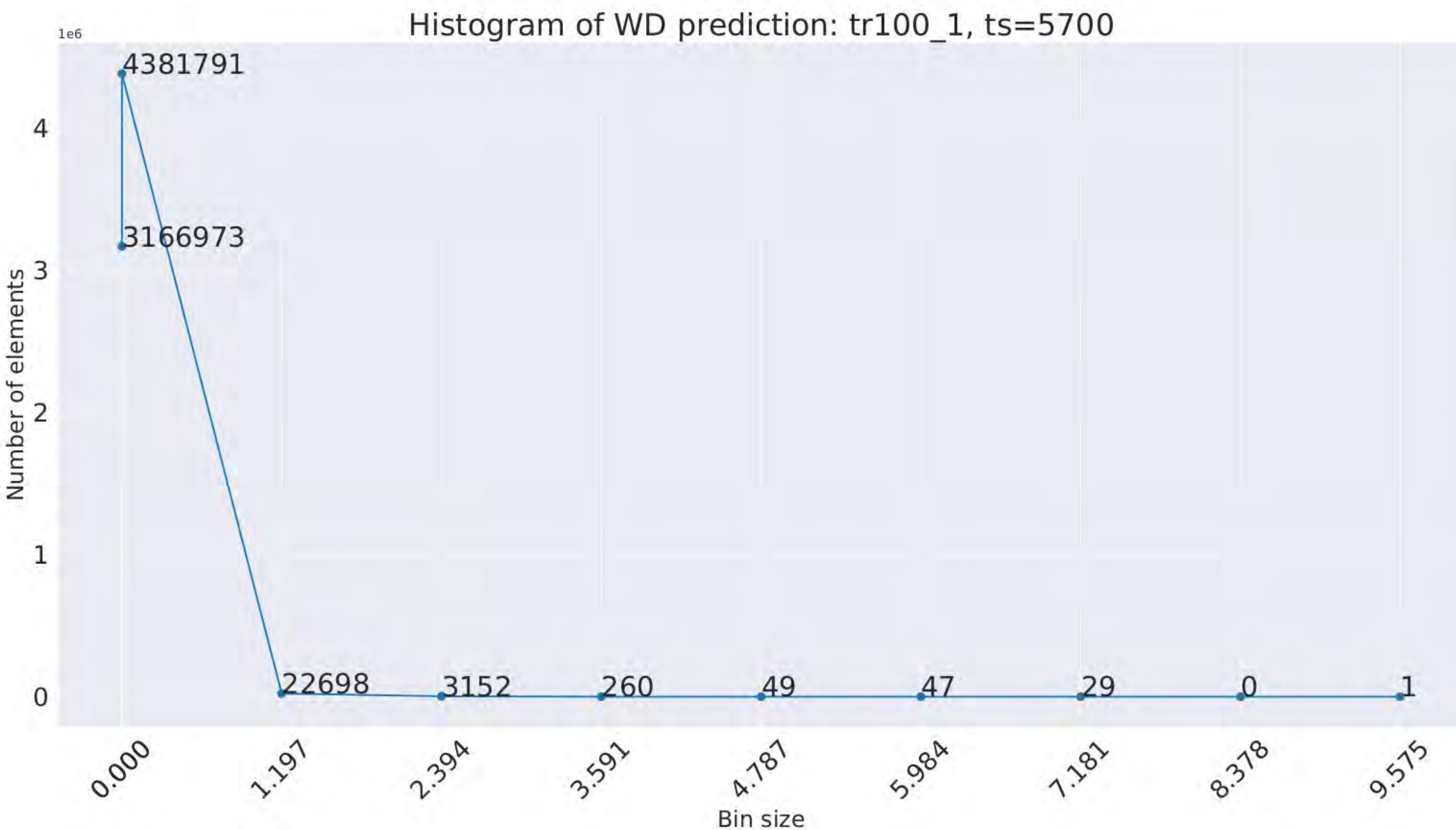


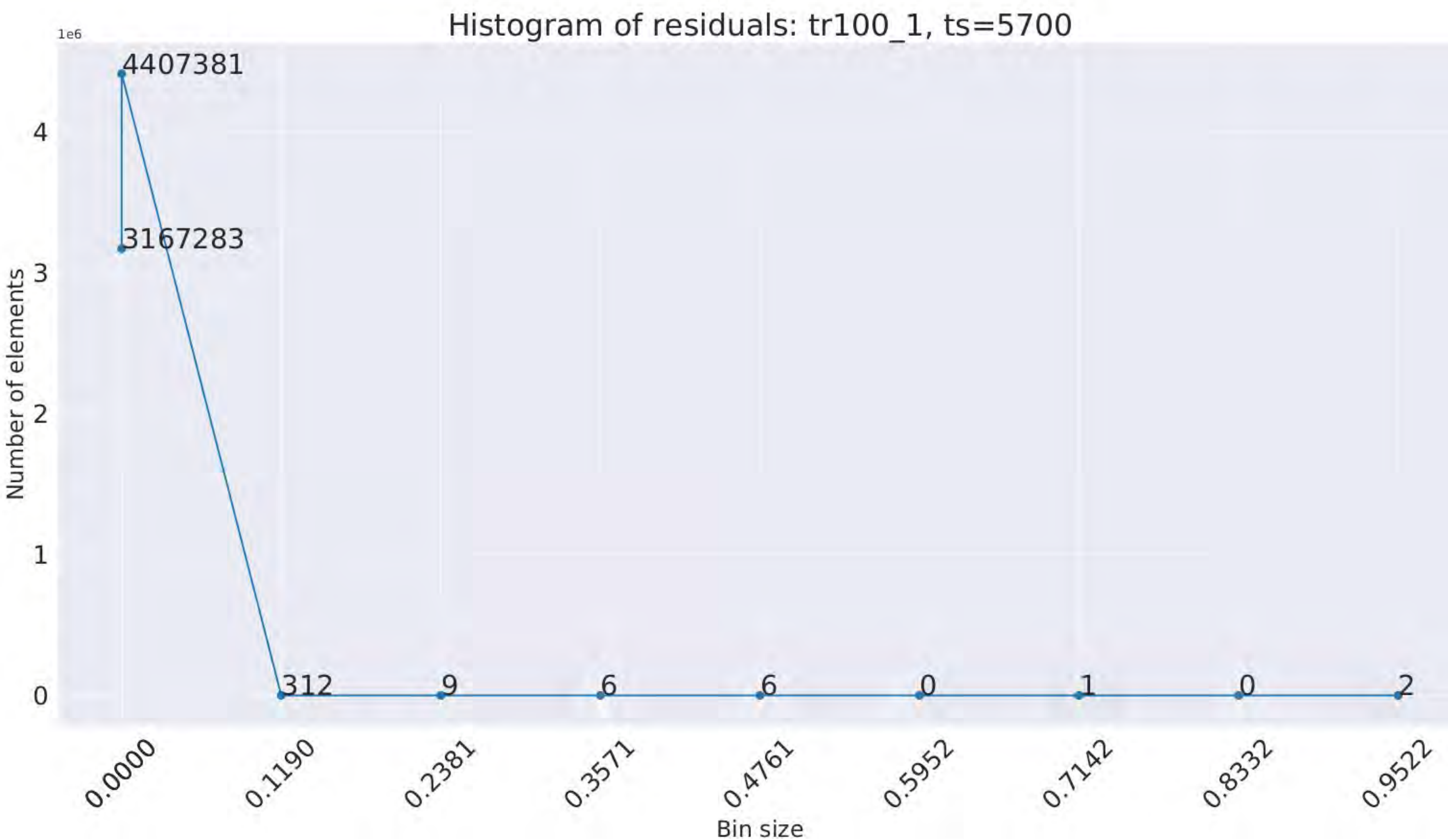
unet, ~75 epochs, tr2_1, t(sec)=2100
gt
predictions
0
1000
2000
0
1000
2000
1.0
0.8
0.6
0.4
0.2
0.0

Boxplot (catchment 709)

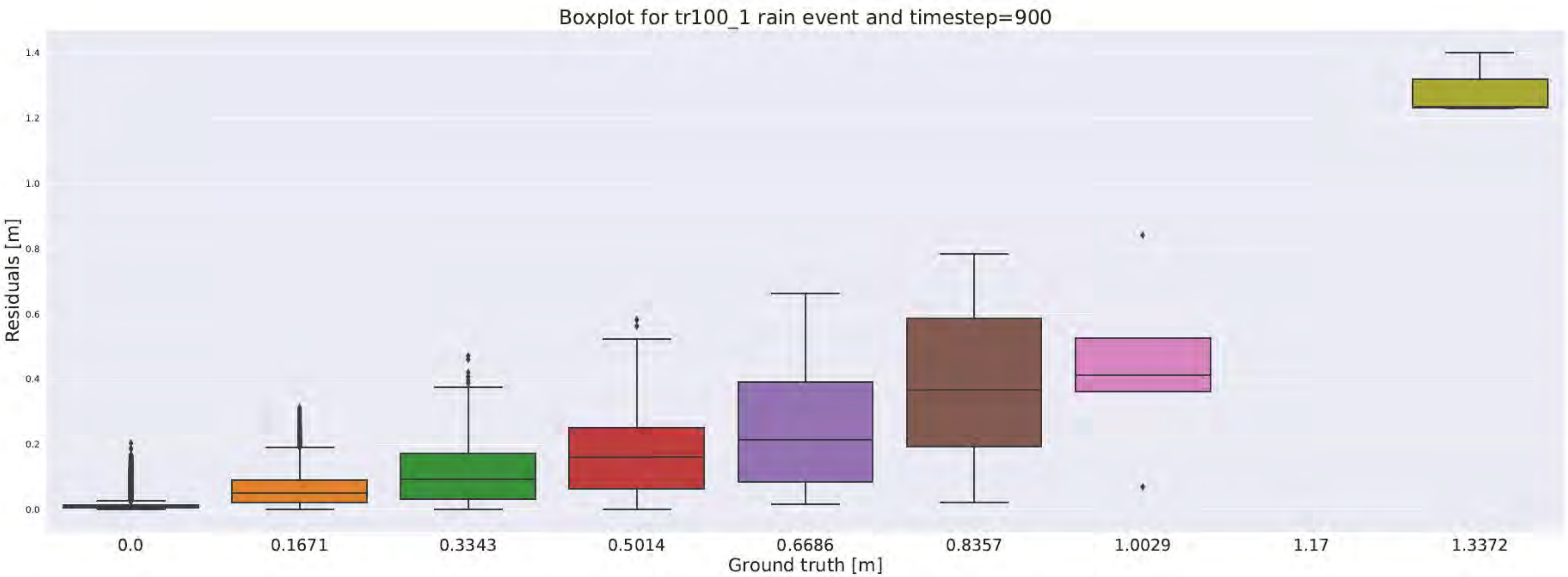


ETH zürich EcoVision eawag aquatic research

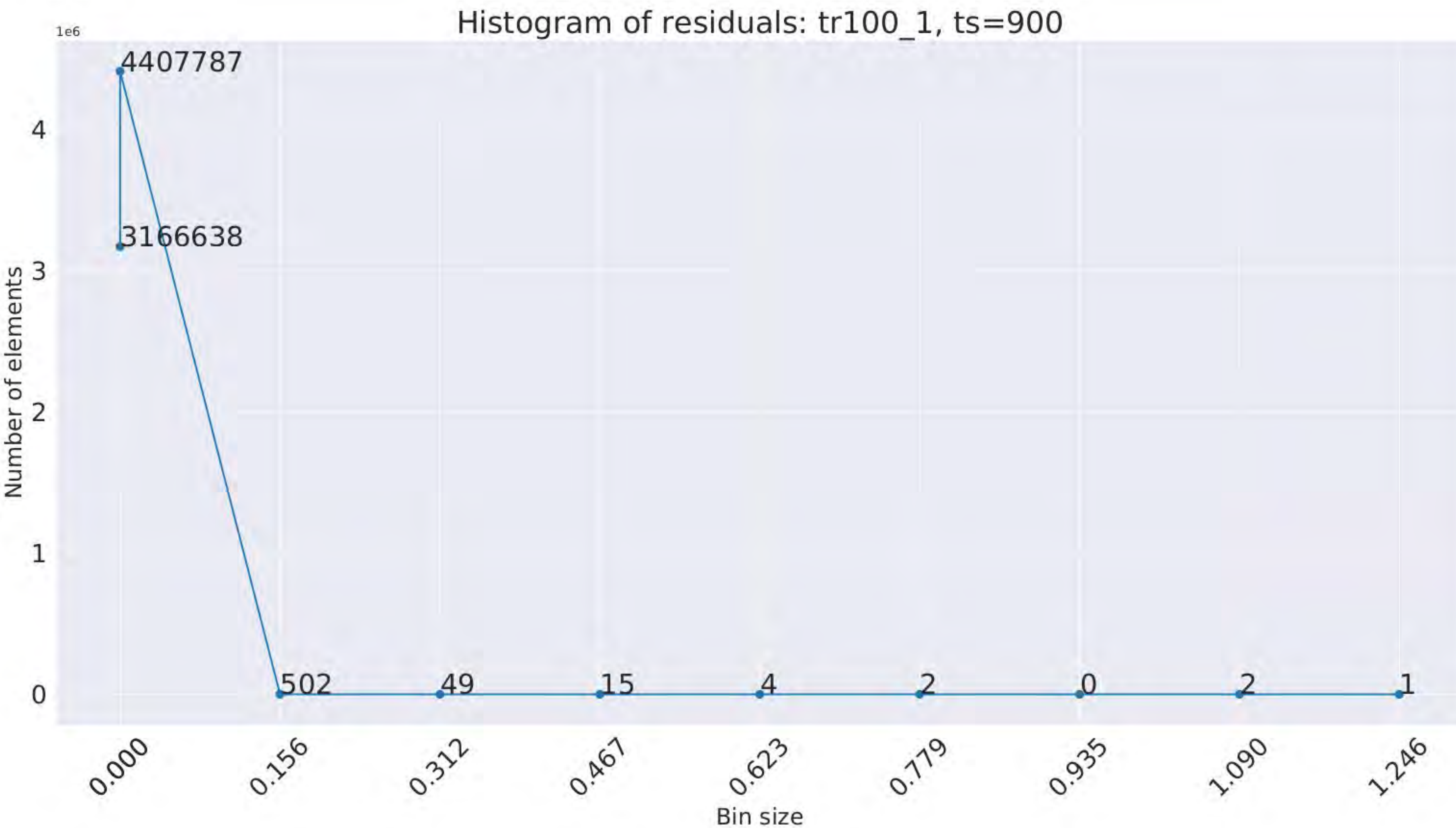



Histogram of residuals: tr100_1, ts=5700
1e6
4407381
3167283
312
9
6
6
0
1
0
2
4
3
2
1
0
Number of elements
0.0000
0.1190
0.2381
0.3571
0.4761
0.5952
0.7142
0.8332
0.9522
Bin size

Boxplot

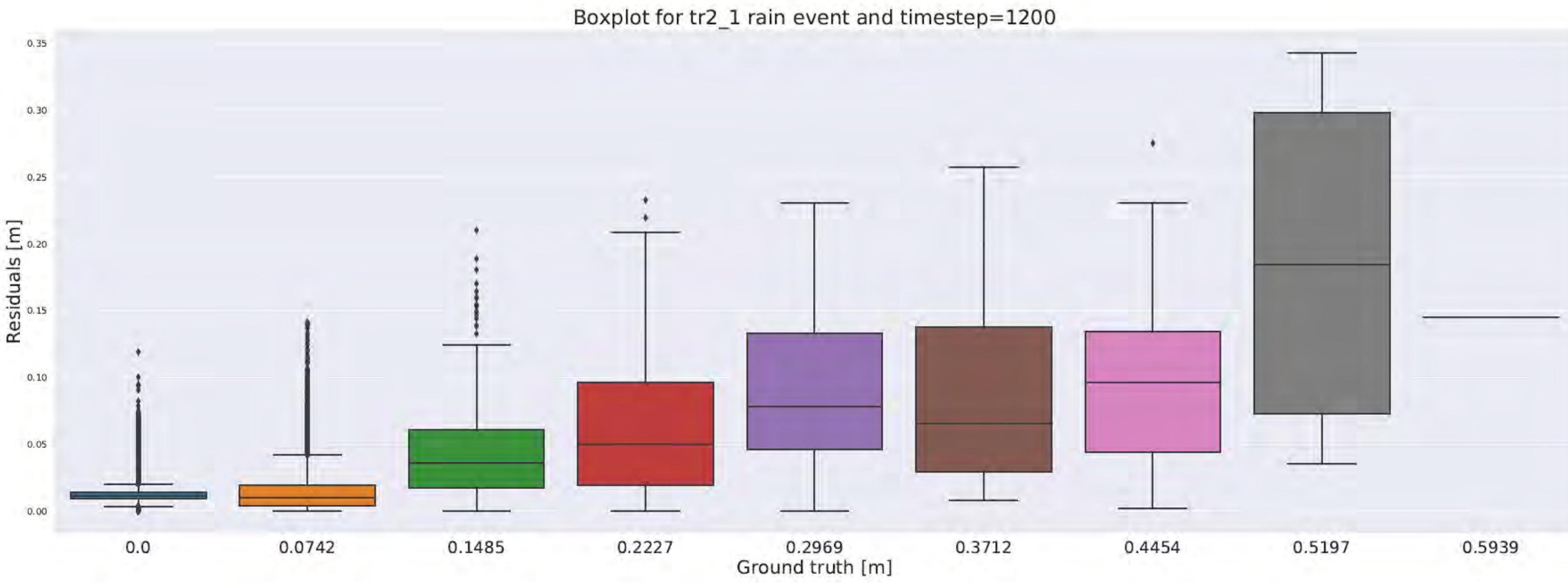


Boxplot for tr100_1 rain event and timestep=900
Residuals [m]
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.1671
0.3343
0.5014
0.6686
0.8357
1.0029
1.17
1.3372
Ground truth [m]



Histogram of residuals: tr100_1, ts=900
1e6
4407787
3166638
502
49
15
4
2
0
2
1
4
3
2
1
0
Number of elements
0.000
0.156
0.312
0.467
0.623
0.779
0.935
1.090
1.246
Bin size

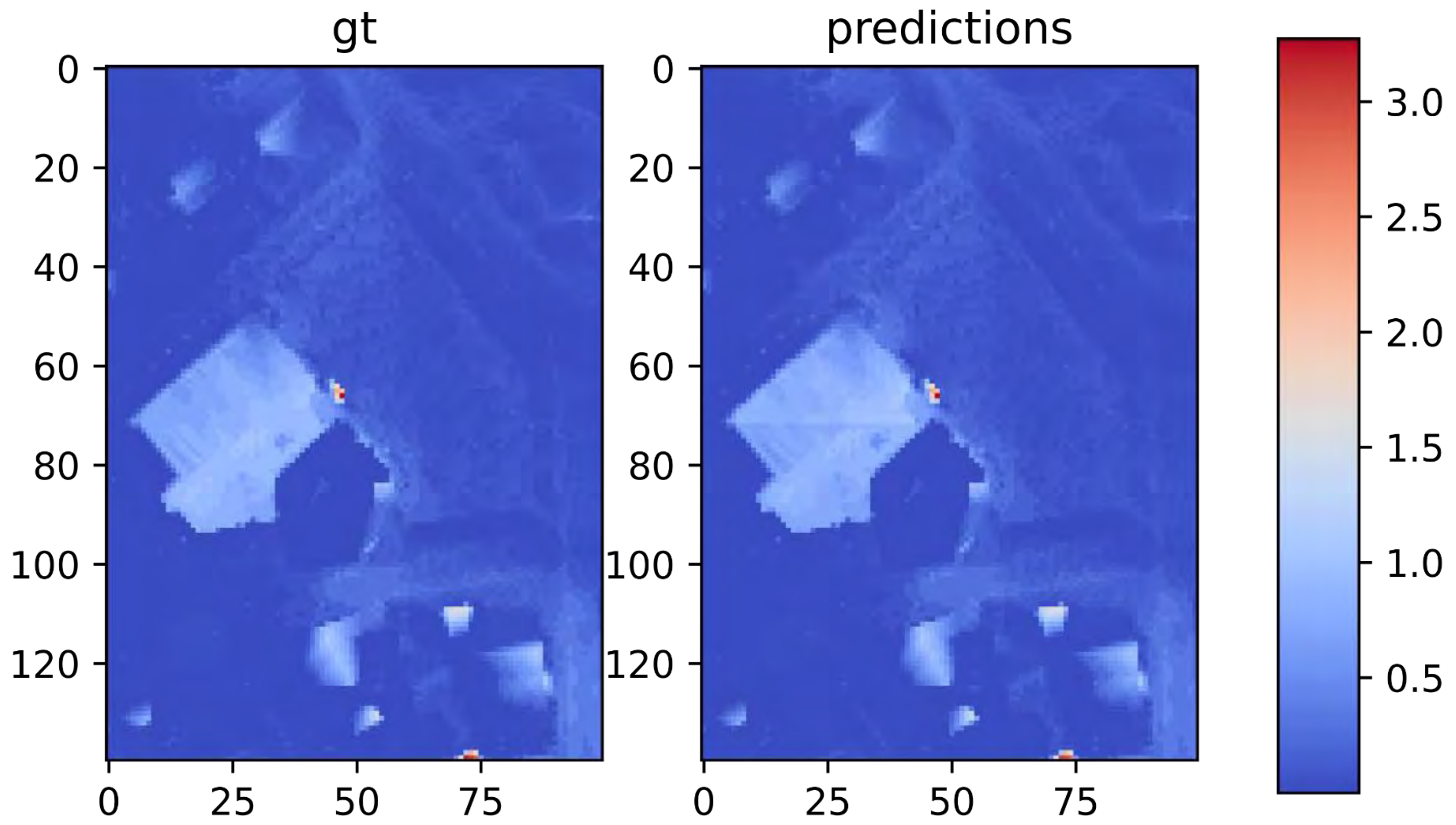
Boxplot


Boxplot for tr2_1 rain event and timestep=1200
Residuals [m]
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.0
0.0742
0.1485
0.2227
0.2969
0.3712
0.4454
0.5197
0.5939
Ground truth [m]

Can you put here a full image of the catchmetnt?

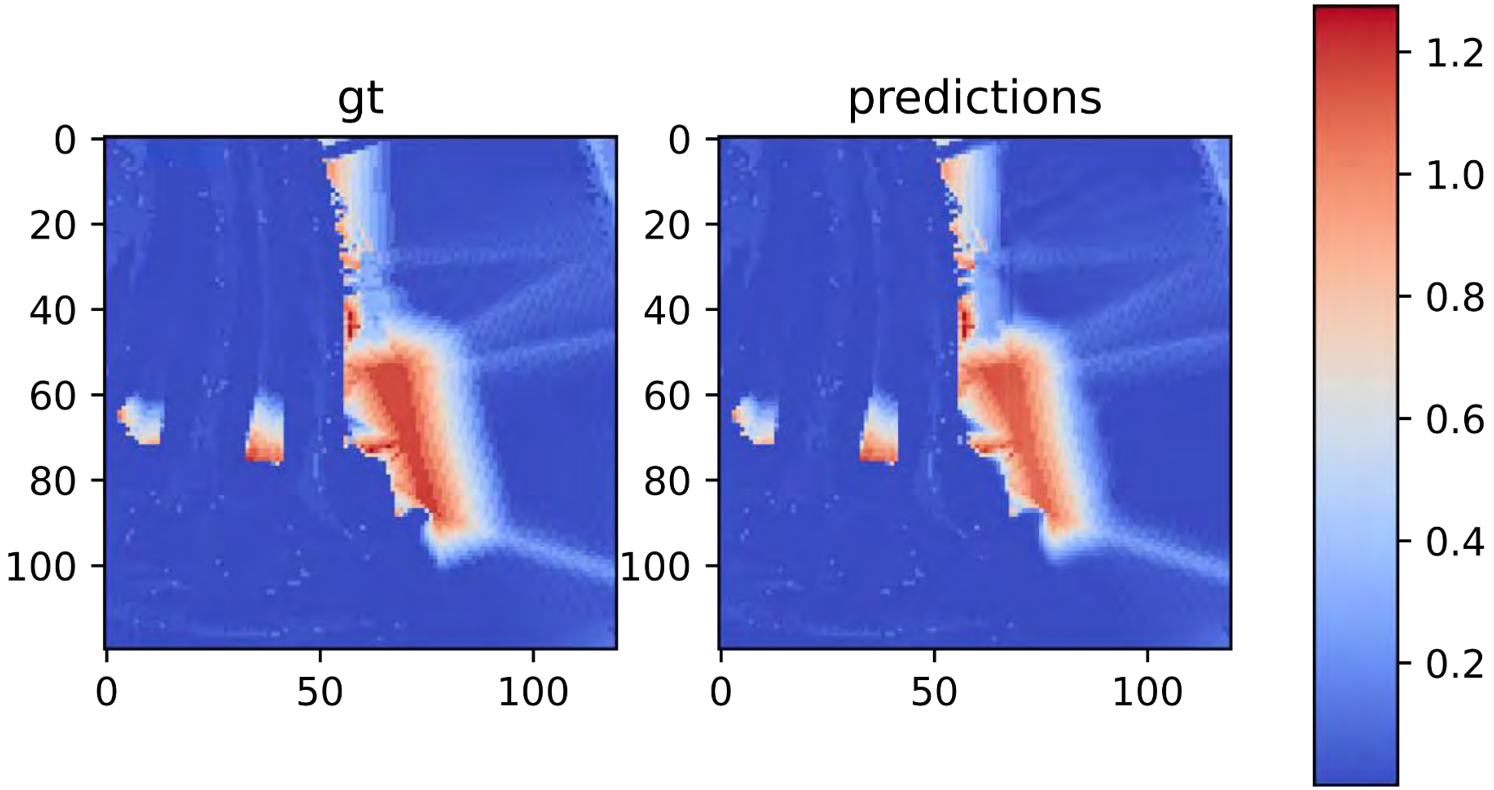
Highlight the areas that you are displaying in the following slides

Patches



ETH zürich | EcoVision | eawag aquatic research

Patches



Performance evaluation

- MAE
  - acceptable MAE error? 5-10% of groundtruth
- Predicting 5-6 timesteps ahead?

Next Steps:

- Bayesian DL
- Add more data? e.g. new catchments
- Implement conv_lstm (multi step ahead prediction)
  - This should already beat the baseline ….

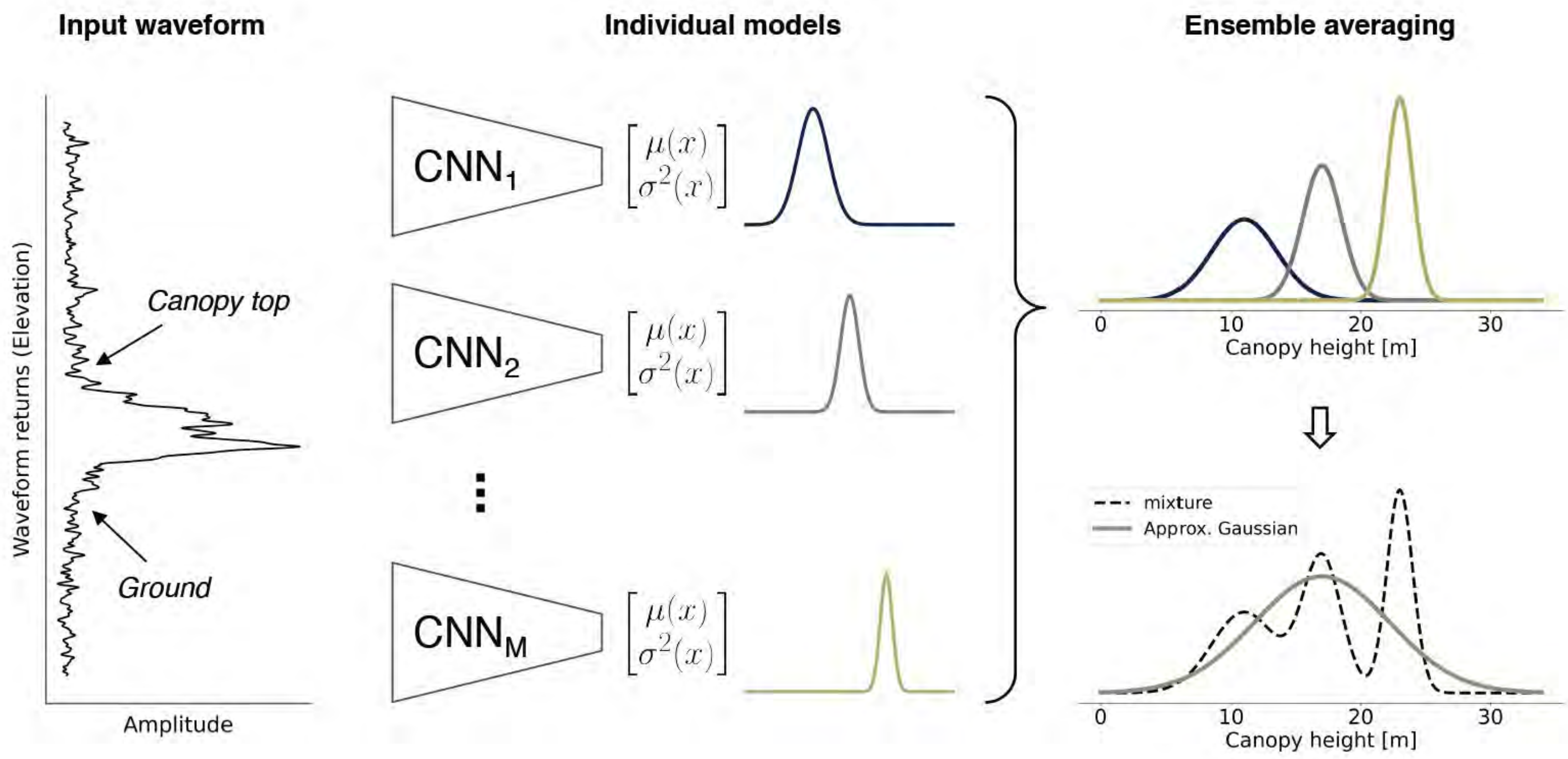
Bayesian DL

Motivation:

- Get well-calibrated uncertainty outputs per pixel for the flood model
- Understand where the model makes trustworthy predictions
- Instead of point prediction output, we predict a distribution over the output to approximate the conditional distribution.

References

- *What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?, NIPS, 2017*
- *Global canopy height estimation with GEDI LIDAR waveforms and Bayesian deep learning, 2021*



Global canopy height estimation with GEDI LIDAR waveforms and Bayesian deep learning, 2021

Gaussian negative log likelihood (NLL)

$$-\log p_\theta(y_n|\mathbf{x}_n) = \frac{\log \sigma_\theta^2(\mathbf{x})}{2} + \frac{\big(y - \mu_\theta(\mathbf{x})\big)^2}{2\sigma_\theta^2(\mathbf{x})} + \text{constant.}$$

Predictive Uncertainty

epistemic uncertainty

aleatoric uncertainty

$$Var(\hat{y}) = \boxed{\frac{1}{M}\sum_{m=1}^{M}\hat{\mu}_m^2 - \left(\frac{1}{M}\sum_{m=1}^{M}\hat{\mu}_m\right)^2} + \boxed{\frac{1}{M}\sum_{m=1}^{M}\hat{\sigma}_m^2},$$

Implementation of variance output

- They enforced the positivity constraint on the variance by passing the second output through the softplus function
- log(1 + exp(.)), and add a minimum variance (e.g. 1e-6) for numerical stability.

Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles, NIPS, 2017

Numerically stable

$$\mathcal{L}_{BNN}(\theta) = \frac{1}{D}\sum_i \frac{1}{2}\exp(-s_i)||\mathbf{y}_i - \hat{\mathbf{y}}_i||^2 + \frac{1}{2}s_i.$$

- In practice, we train the network to predict the log variance.
- it is more numerically stable than regressing the variance, σ^2, as the loss avoids a potential division by zero.

What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?, NIPS, 2017

```
_________________________________________________________________
Layer (type)                 Output Shape
=================================================================
conv2d_1(Conv2D)              (None, 120, 120, 6)
_________________________________________________________________
activation_1 (Activation)      (None, 120, 120, 6)
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 120, 120, 16)
_________________________________________________________________
activation_2 (Activation)    (None, 120, 120, 16)
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 120, 120, 32)
_________________________________________________________________
activation_3 (Activation)    (None, 120, 120, 32)
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 120, 120, 64)
_________________________________________________________________
activation_4 (Activation)    (None, 120, 120, 64)
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 120, 120, 128)
_________________________________________________________________
activation_5 (Activation)    (None, 120, 120, 128)
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 120, 120, 1)

=================================================================
```

Experiments

- One timestep experiments
  - input: 3 channels, (n x n x 3)
  - output: 1 channel (n x n x1)
- Data: toy
- Models:
  - Basic conv Net (Tc_net)
  - Unet
  - Uresnet
  - Resnet
- Performance evaluation:
  - MAE (mean MAE across the catchment, at each time step)
  - acceptable MAE error: 5-10% WD

DEM	$water\_depth_{(t1)}$	dem $rainfall_{(t1)}$

Toy catchment

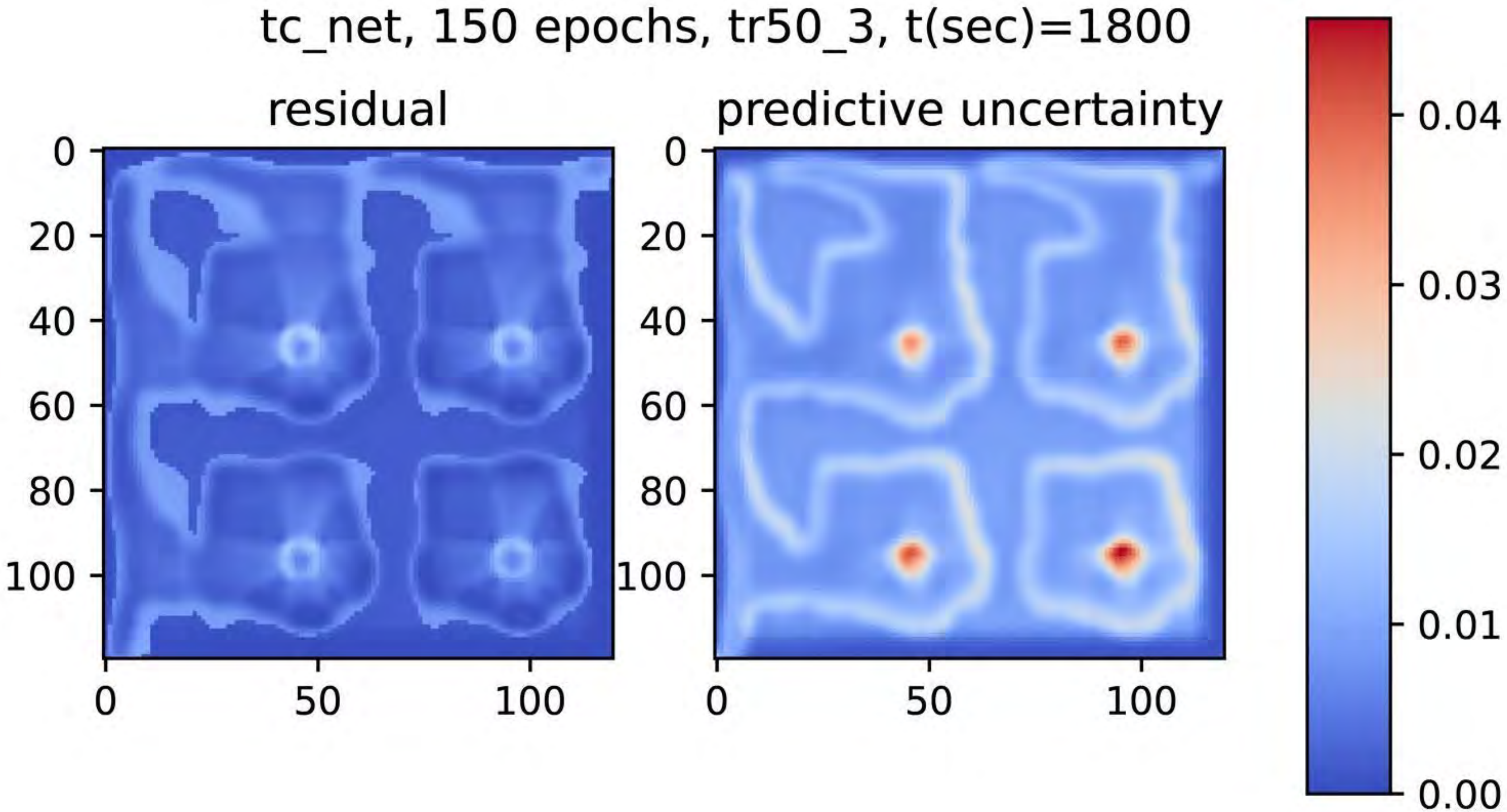
Experimen t	tr2 (MAE m)	tr10 (MAE m)	tr100_1 (MAE m)	tr5_2 (MAE m)	tr20_3 (MAE m)	tr50_3 (MAE m)	Average (MAE m)
UNet L1	0.0091	0.0106	0.0186	0.0129	0.0140	0.0178	**0.0139**
tc_net L1 (lr=1e-4)	0.0108	0.0114	0.0181	0.0126	0.0164	0.0181	0.0146
tc_net L2 (lr=1e-4)	0.0127	0.0139	0.0222	0.0141	0.0182	0.0207	0.0170

Baseline 1timestep	0.0332	0.0535	0.0726	0.0451	0.0593	0.0668	0.0551

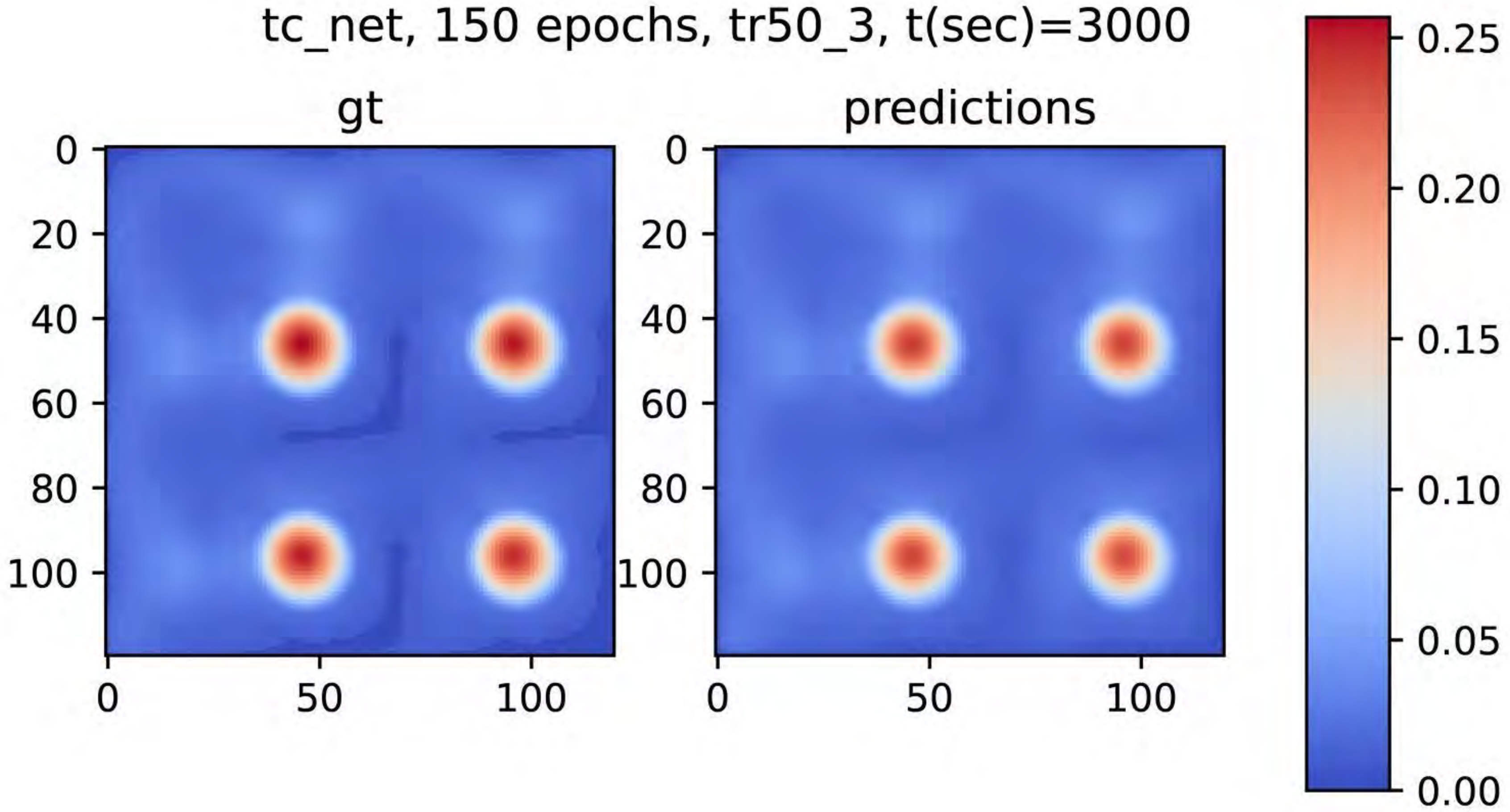
Toy catchment

Experimen t	tr2 (MAE m)	tr10 (MAE m)	tr100_1 (MAE m)	tr5_2 (MAE m)	tr20_3 (MAE m)	tr50_3 (MAE m)	Average (MAE m)
tc_net **bay** (lr=1e-4)	0.0188	0.0220	0.0372	0.0209	0.0280	0.0331	0.0267
tc_net L1 (lr=1e-4)	0.0108	0.0114	0.0181	0.0126	0.0164	0.0181	**0.0146**
tc_net L2 (lr=1e-4)	0.0127	0.0139	0.0222	0.0141	0.0182	0.0207	0.0170

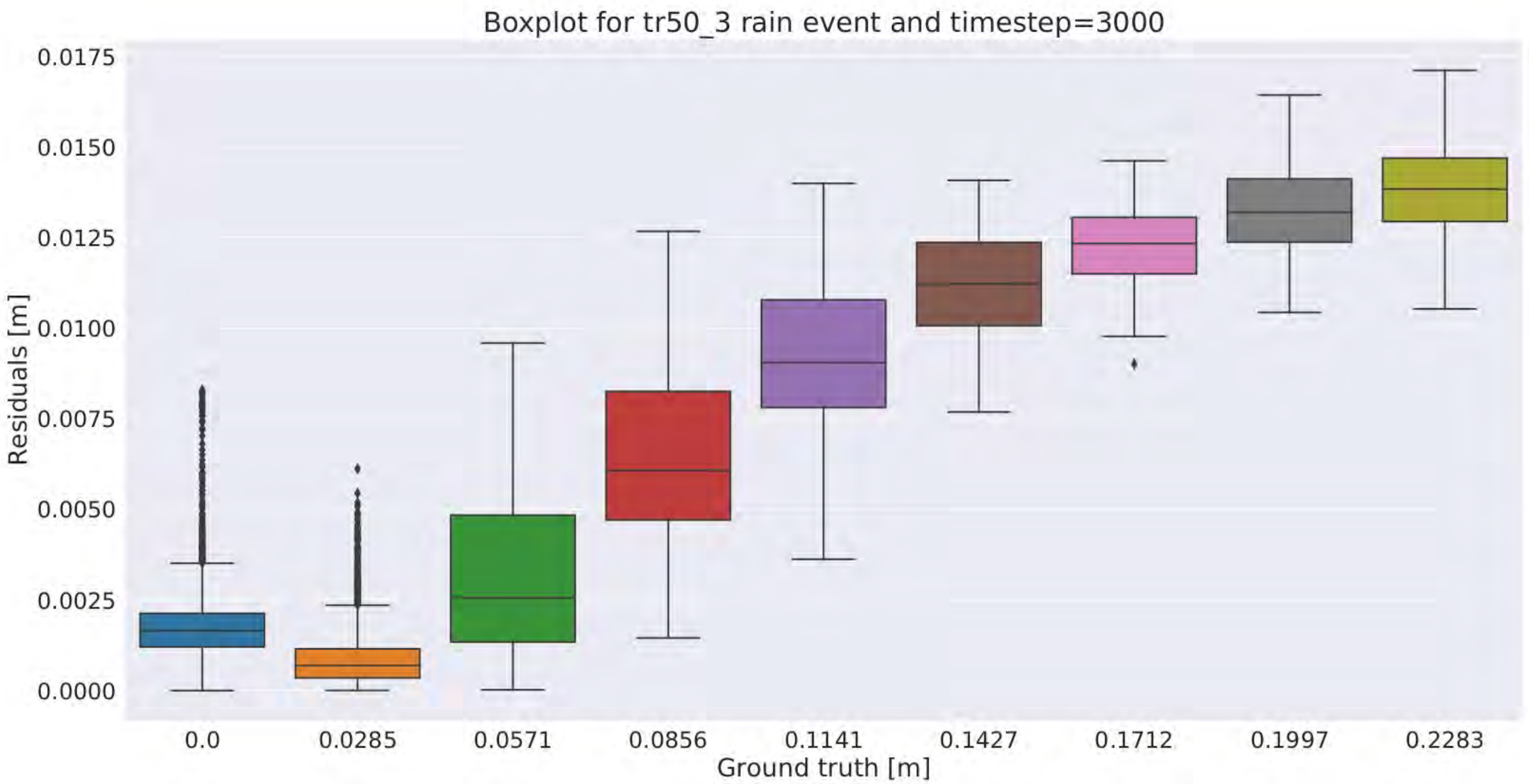
Baseline 1timestep	0.0332	0.0535	0.0726	0.0451	0.0593	0.0668	0.0551



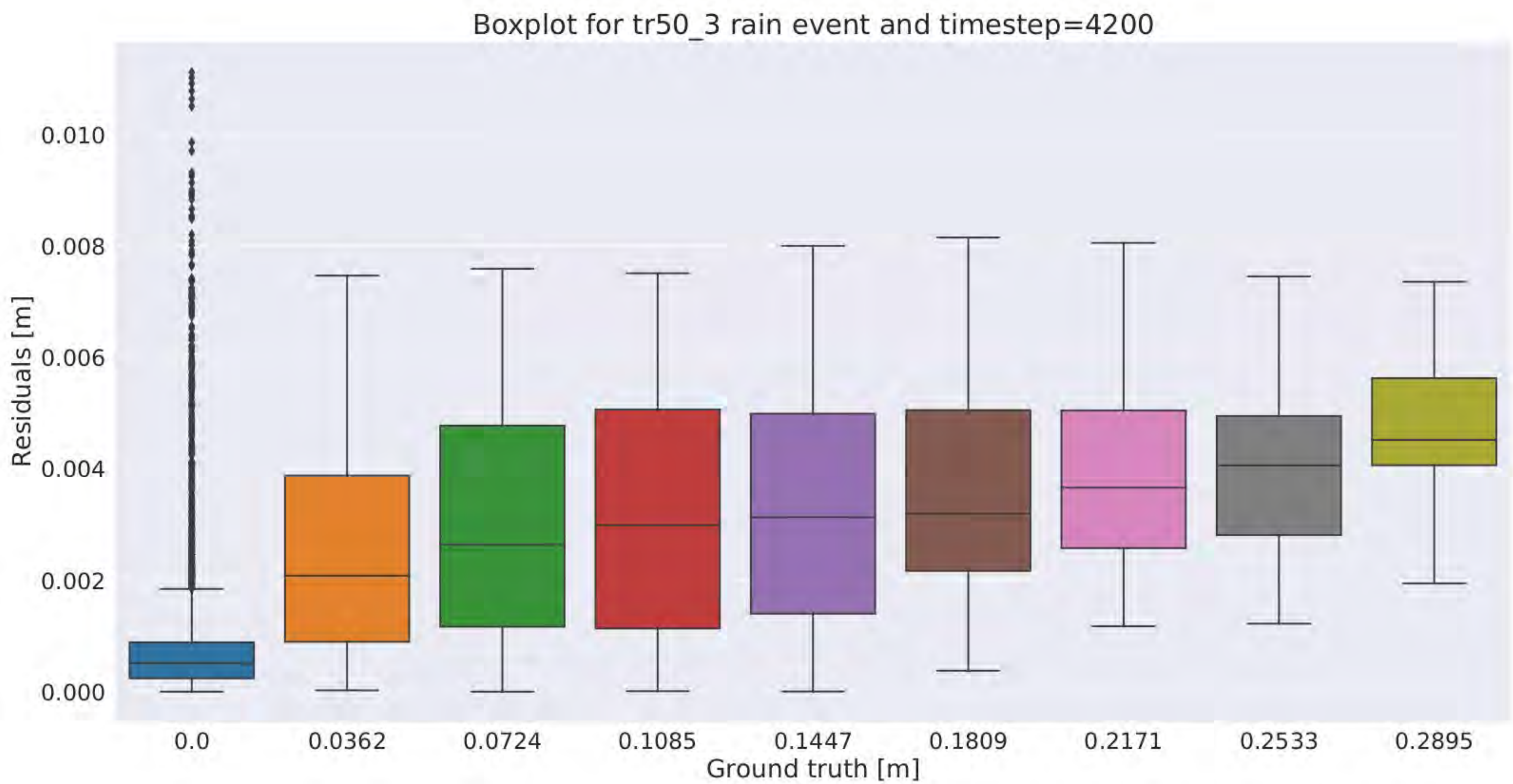
tc_net, 150 epochs, tr50_3, t(sec)=1800
residual
predictive uncertainty
0
20
40
60
80
100
0
50
100
0.04
0.03
0.02
0.01
0.00

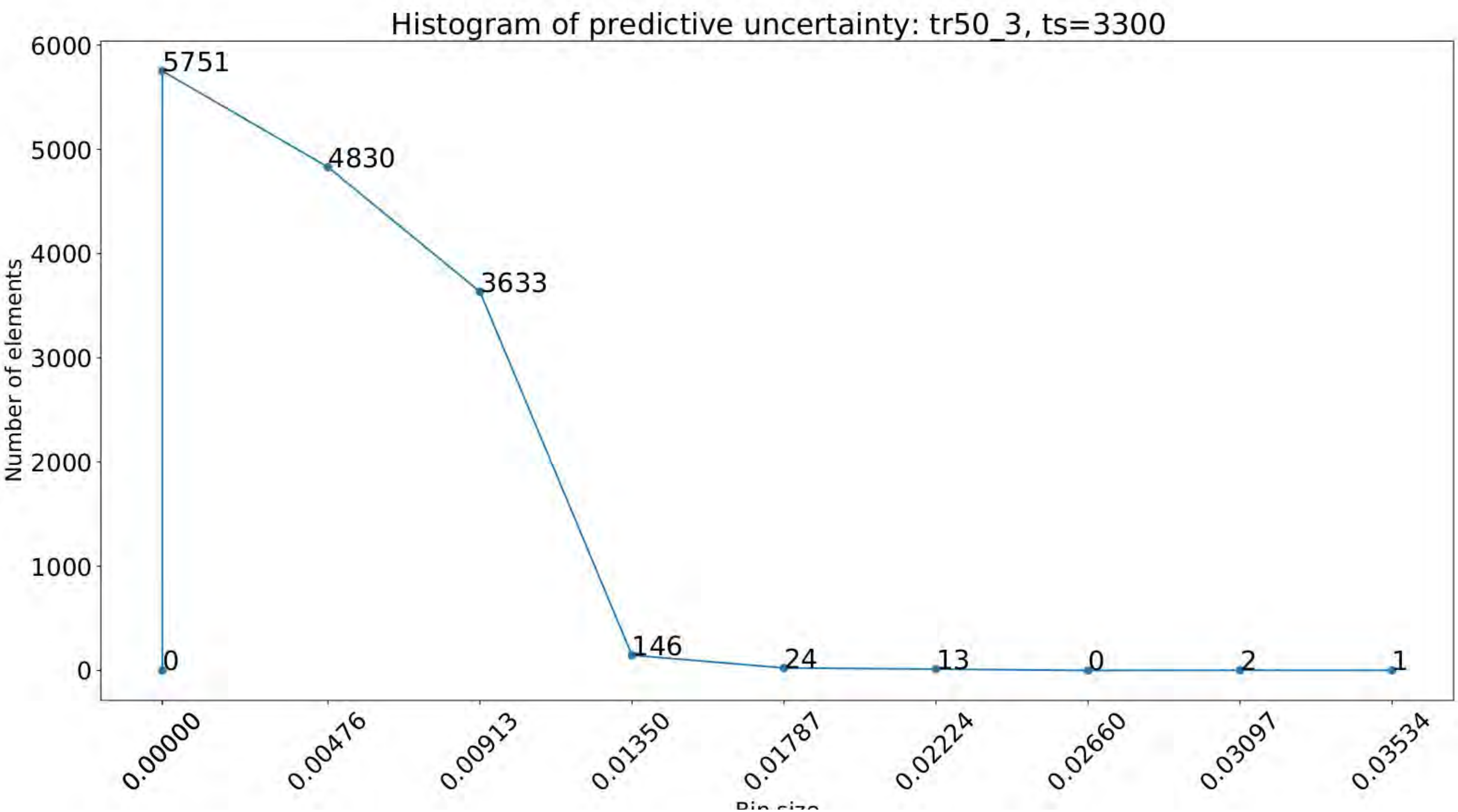

tc_net, 150 epochs, tr50_3, t(sec)=3000
gt
predictions
0
20
40
60
80
100
0
50
100
0.25
0.20
0.15
0.10
0.05
0.00

Box plots of residuals - toy catchment



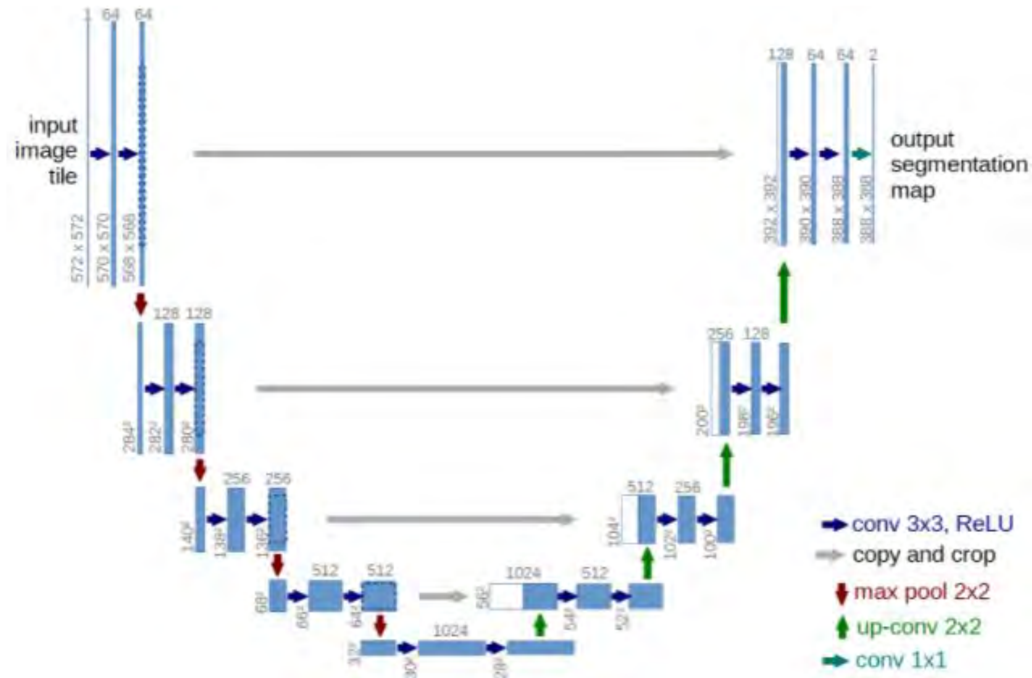
Box plots of residuals - toy catchment



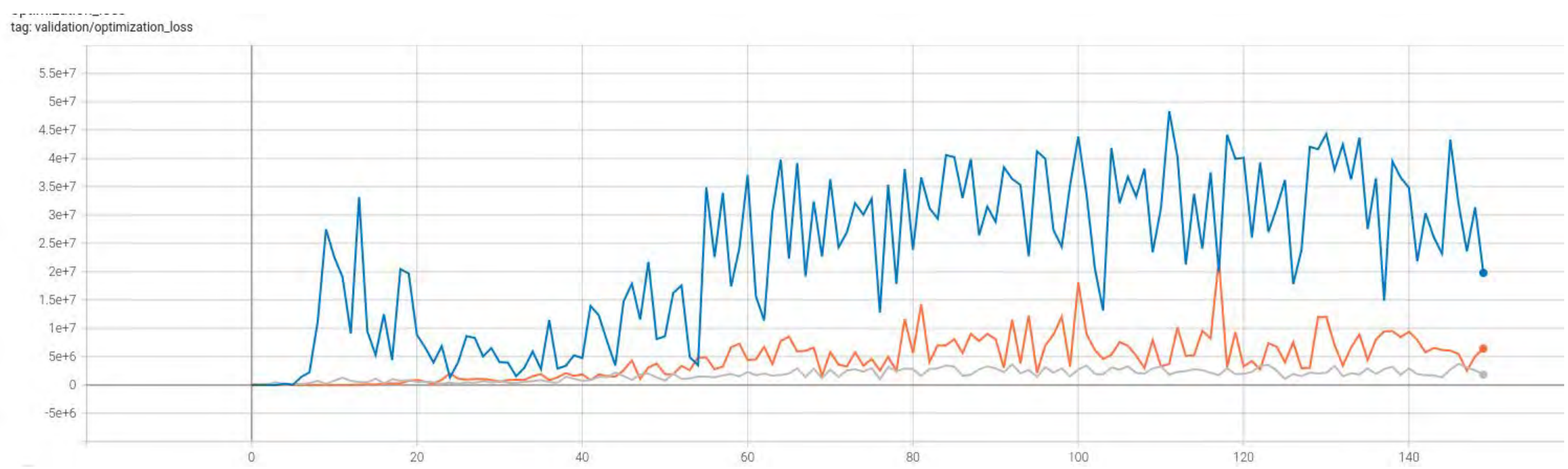

Histogram of predictive uncertainty: tr50_3, ts=3300
Number of elements
6000
5000
4000
3000
2000
1000
0
5751
4830
3633
146
0
24
13
0
2
1
0.00000
0.00476
0.00913
0.01350
0.01787
0.02224
0.02660
0.03097
0.03534

Tests with other models

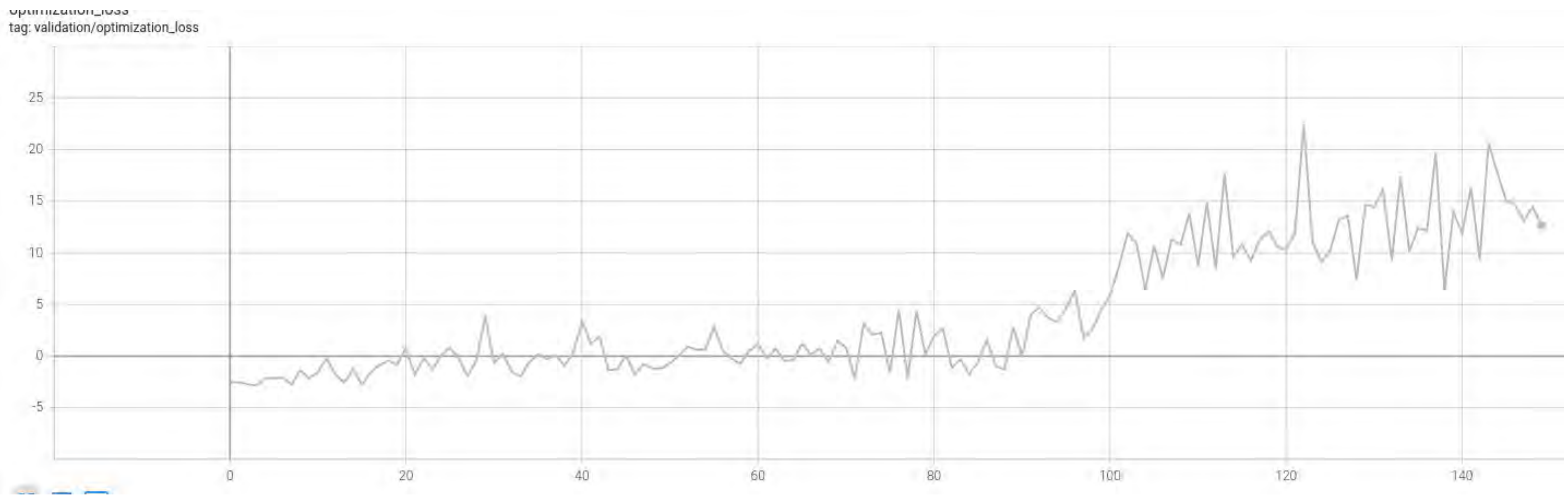
- Unet
- Uresnet
- Resnet



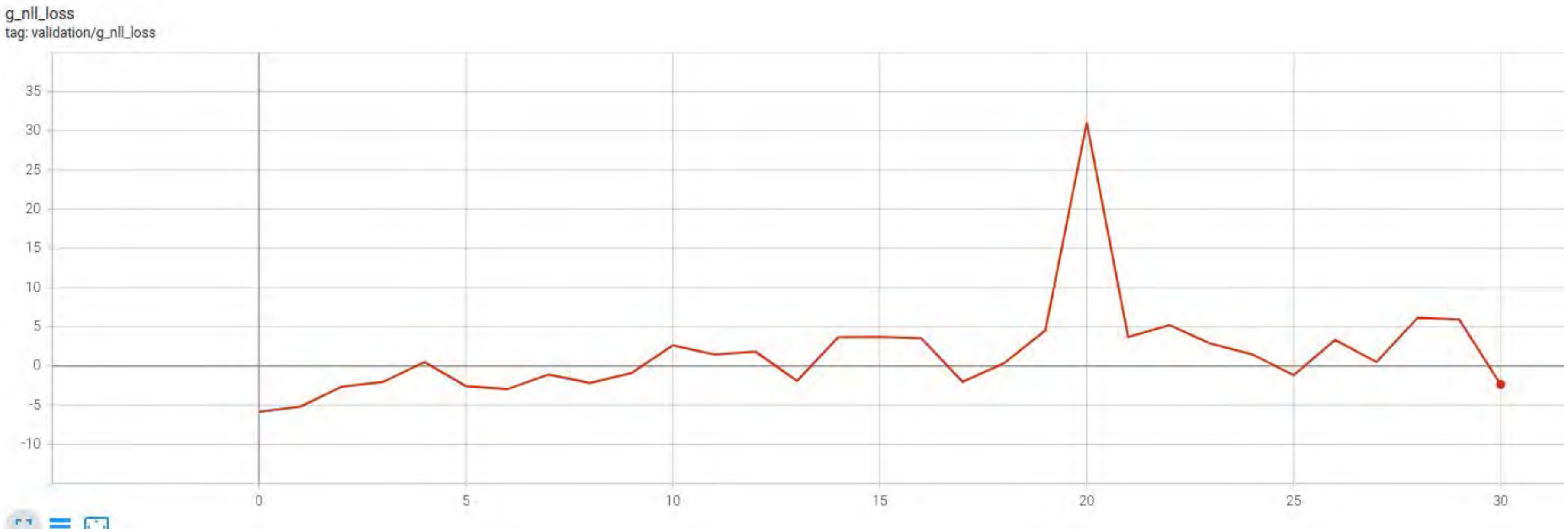
Validation loss curve - UNet (toy)



Validation loss curve - Resnet (toy)



Validation loss curve - Uresnet (709)



Toy catchment - Bayesian DL - Conclusions

- We use Bayesian approach as it directly provides a well-calibrated uncertainty together with every estimate.
- We estimate uncertainty with an ensemble of five separate TcNet models that were trained independently, starting from different random initializations.
- Prediction MAE: 2.67 cms
- UNet, UResNet – not working on toy dataset

Laplacian Negative Log Likelihood

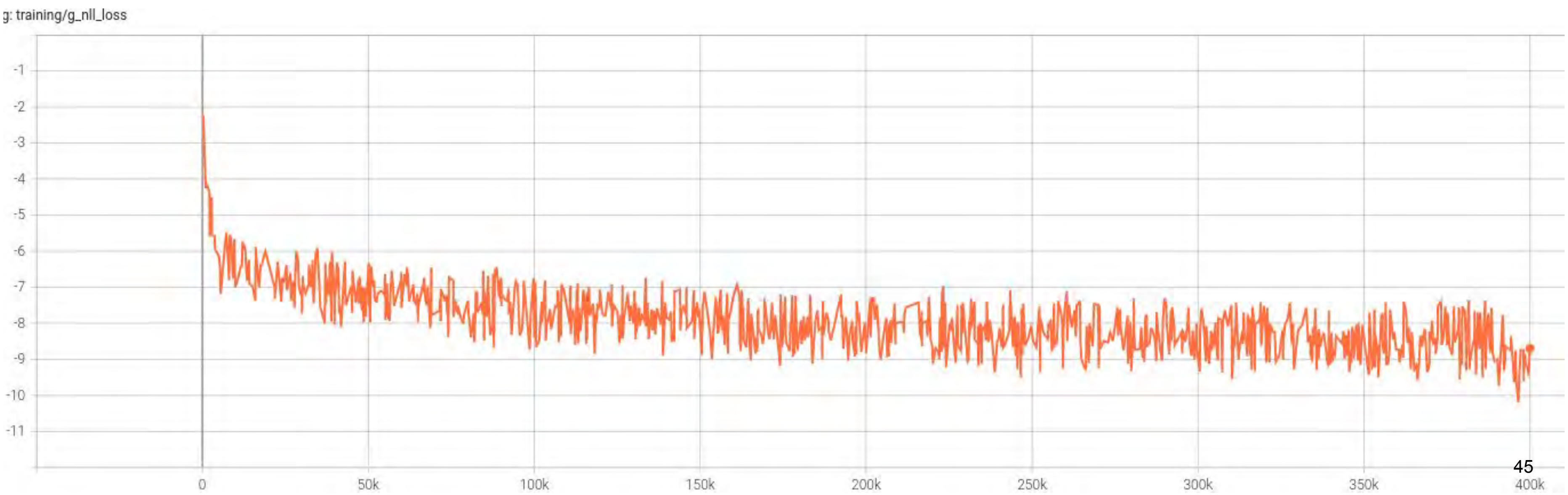
- By assuming the noise is Laplacian, the negative log-likelihood to be minimized is:

$$-\log p(y|\tilde{y}, \sigma) = \frac{|y - \tilde{y}|}{\sigma} + \log \sigma + const.$$

- Key idea is to predict a posterior probability distribution for each pixel parameterized with its mean as well as its variance $p(y|\tilde{y}, \sigma)$ over ground-truth labels y.

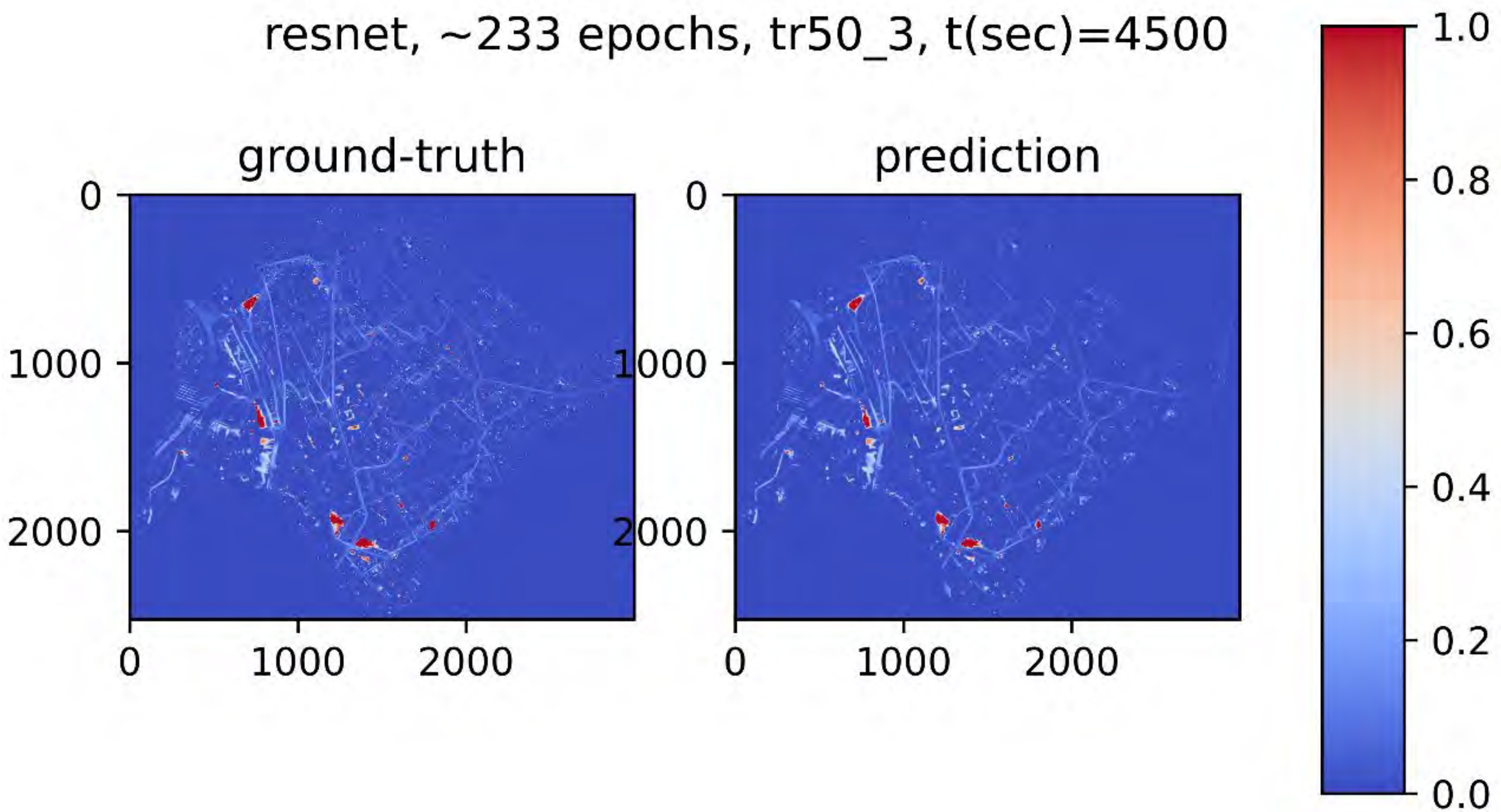
D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry, CVPR, 2020

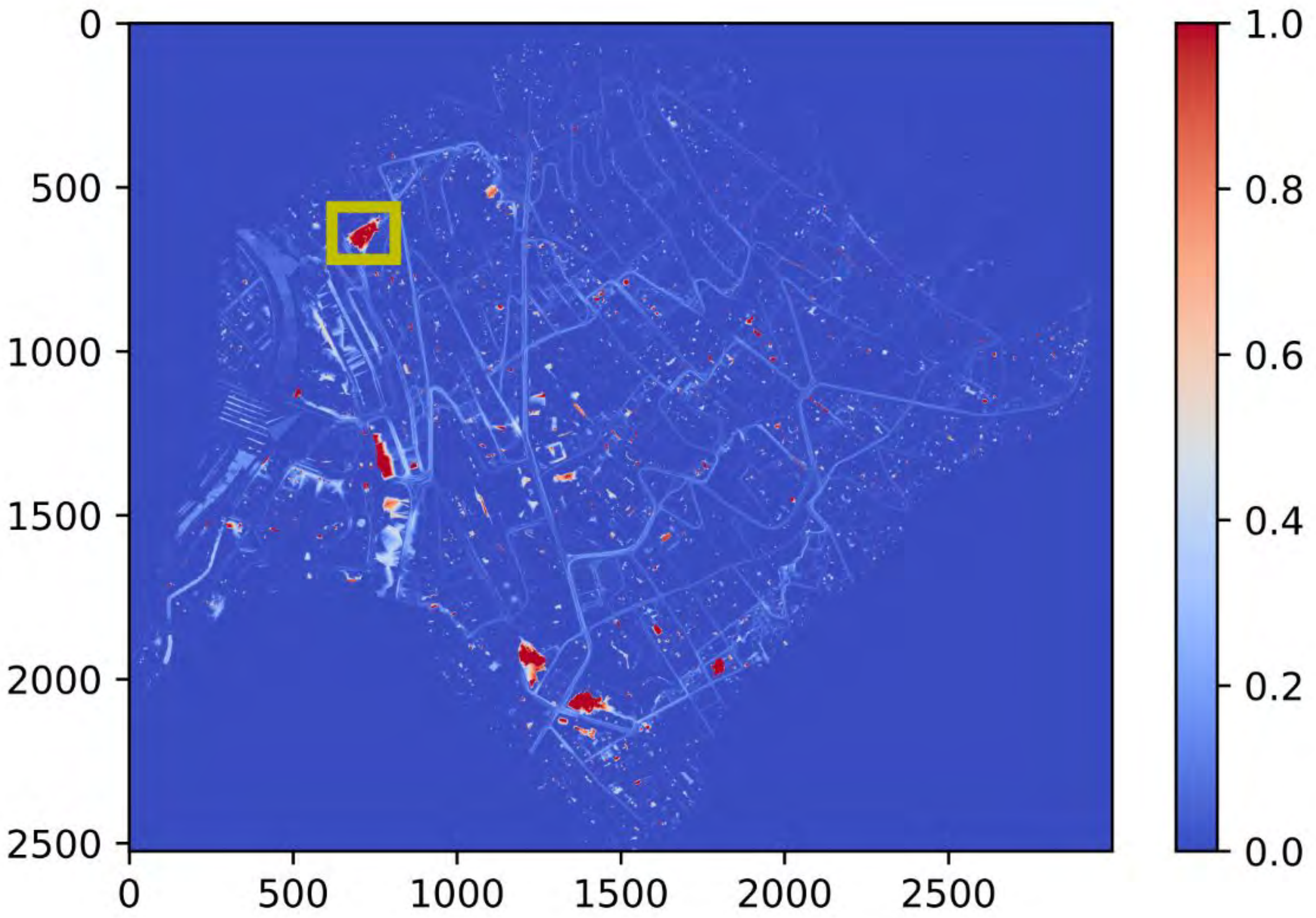
Training curve



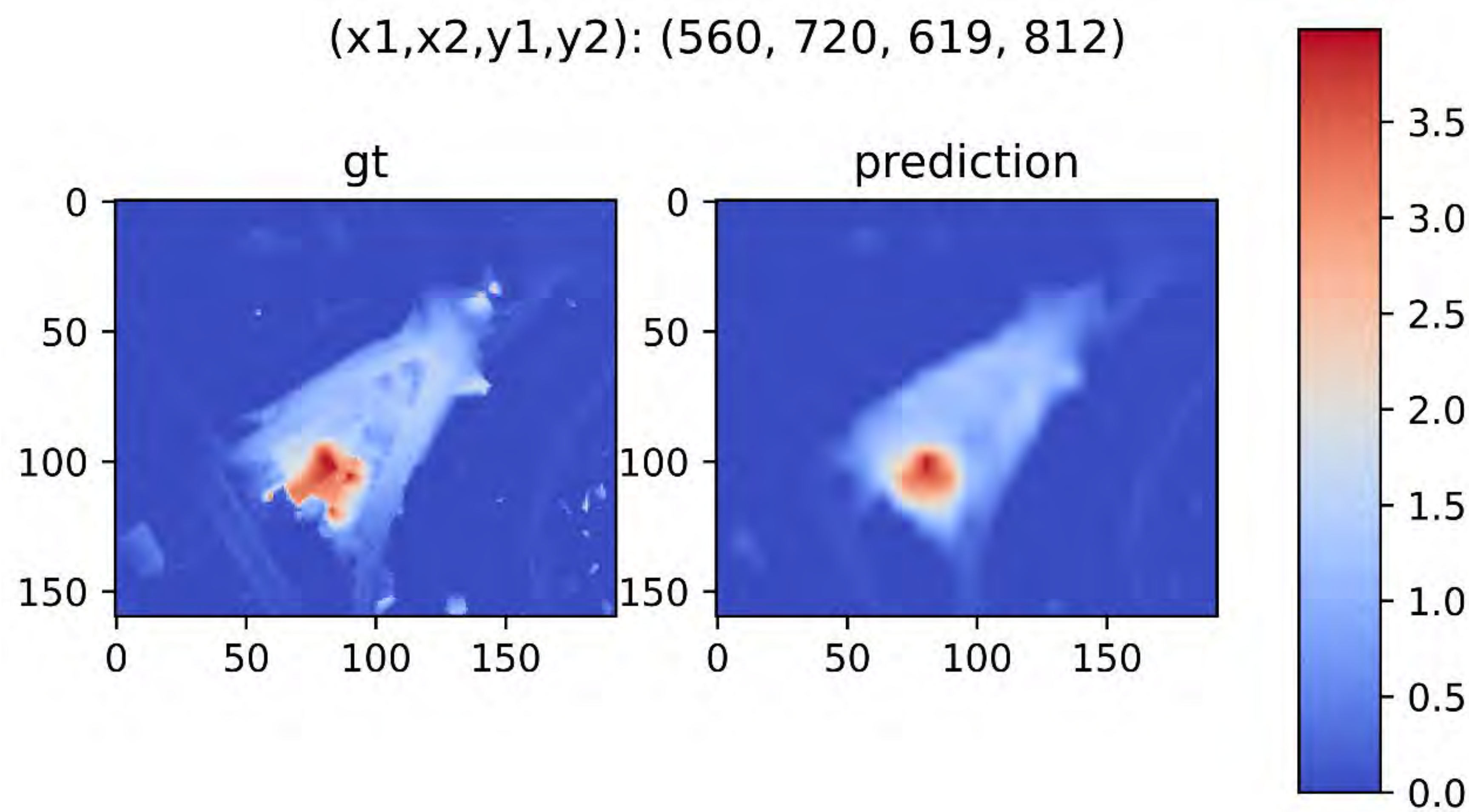
Validation curve

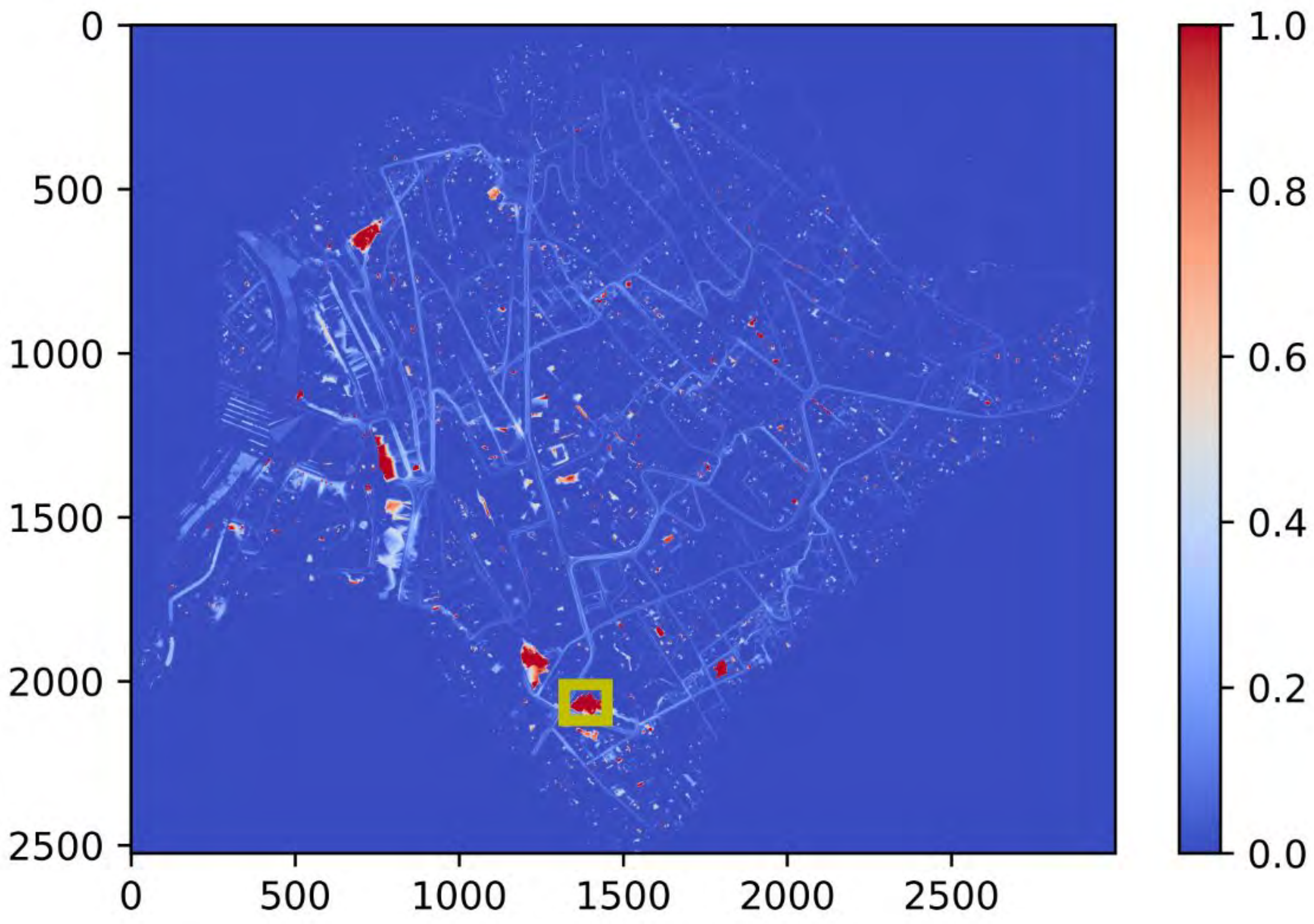
Gt vs Pred, tr50_3, ts=4500



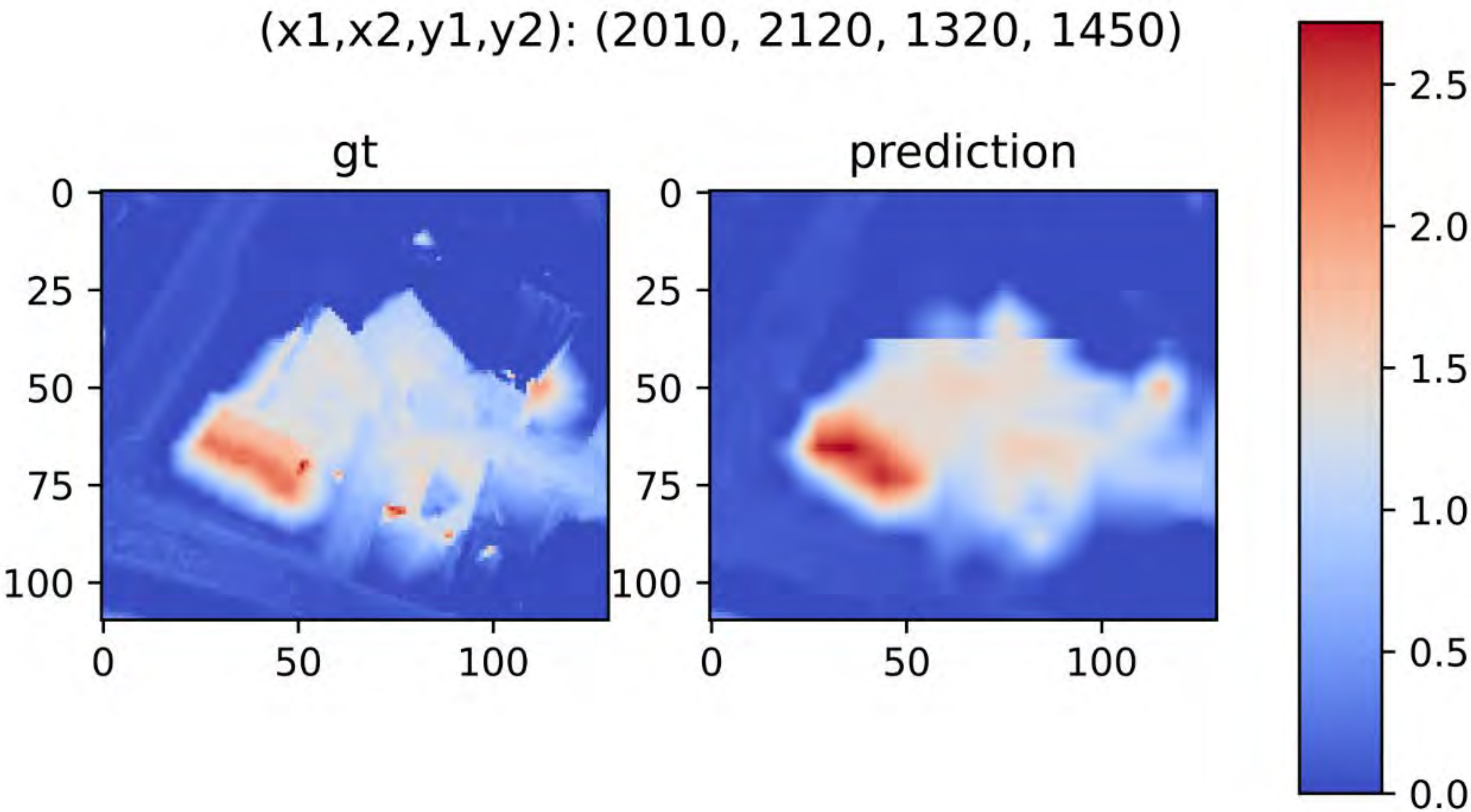

0
500
1000
1500
2000
2500
0
500
1000
1500
2000
2500
1.0
0.8
0.6
0.4
0.2
0.0

Gt vs Pred, tr50_3, ts=4500, patches

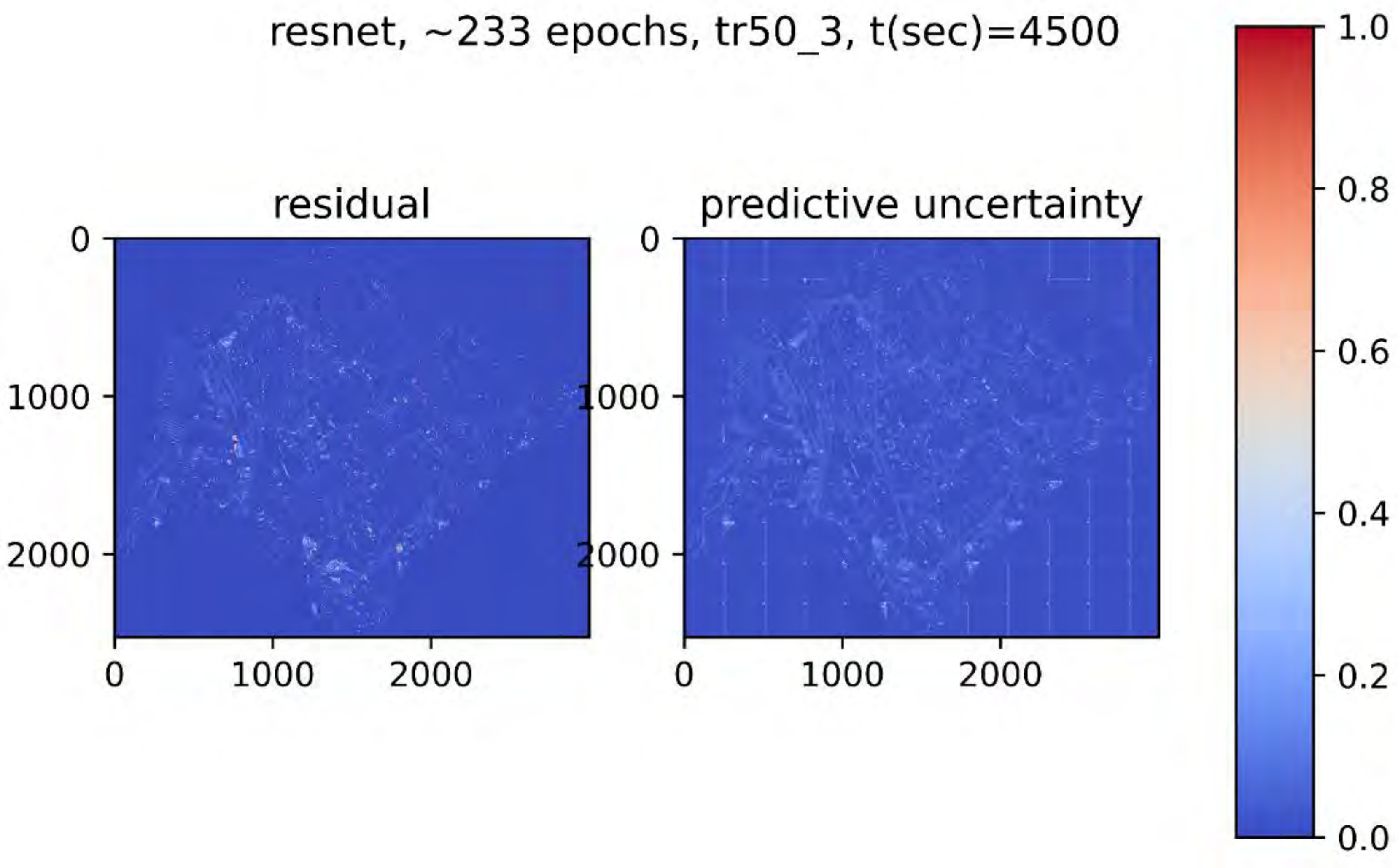



0
500
1000
1500
2000
2500
0
500
1000
1500
2000
2500
1.0
0.8
0.6
0.4
0.2
0.0

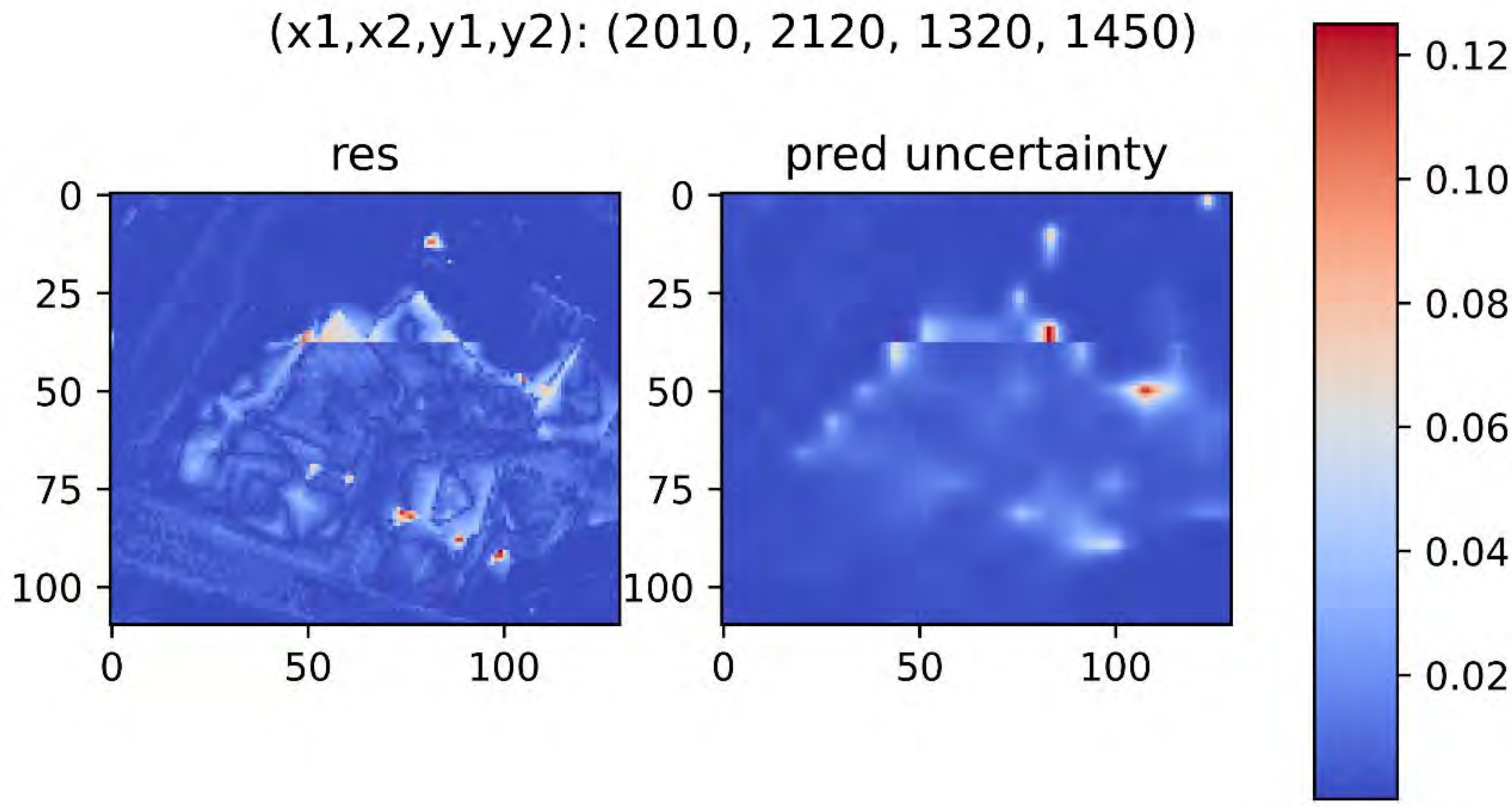
Gt vs Pred, tr50_3, ts=4500, patches



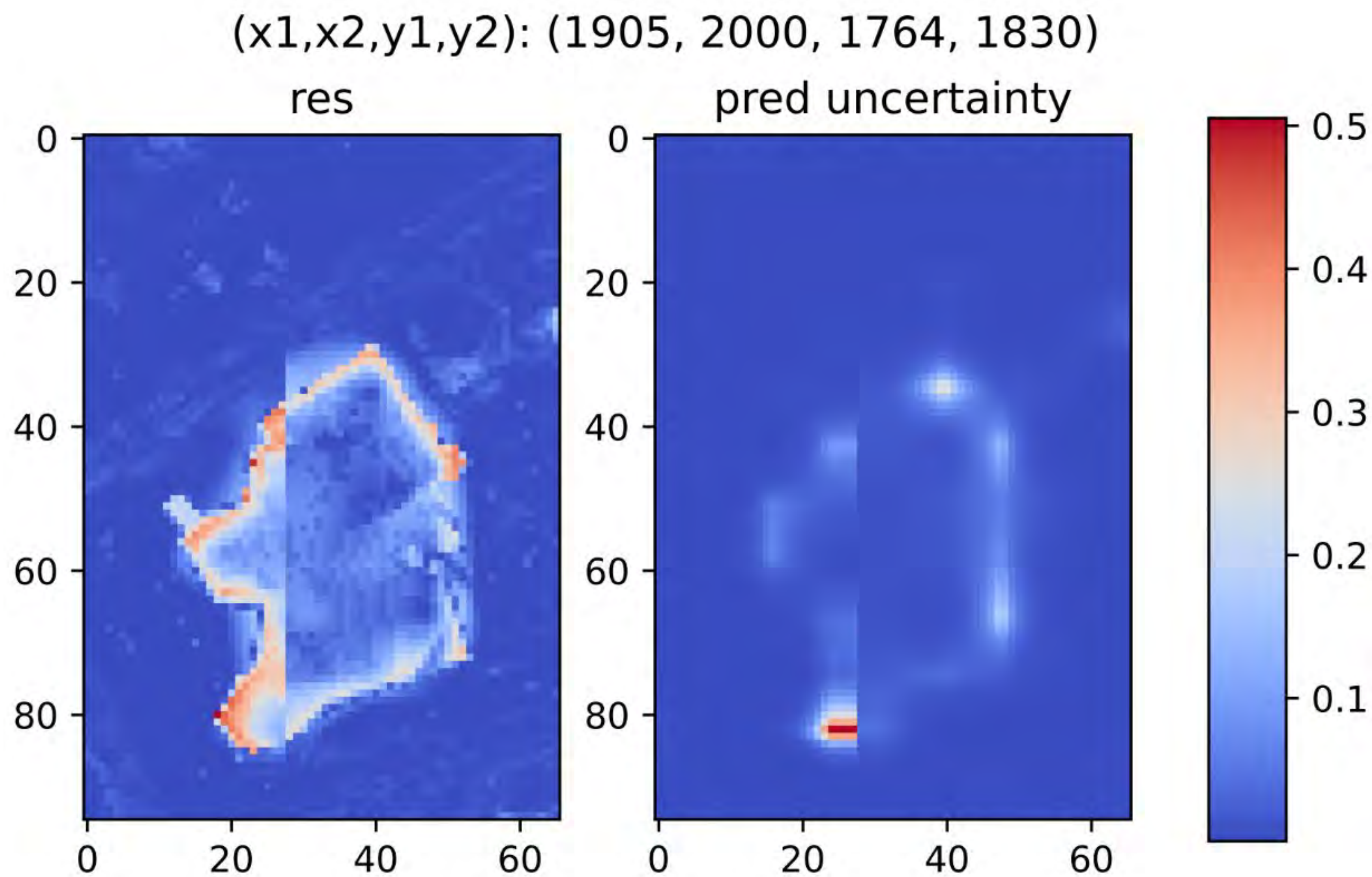
Res vs Pred uncertainty, tr50_3, ts=4500



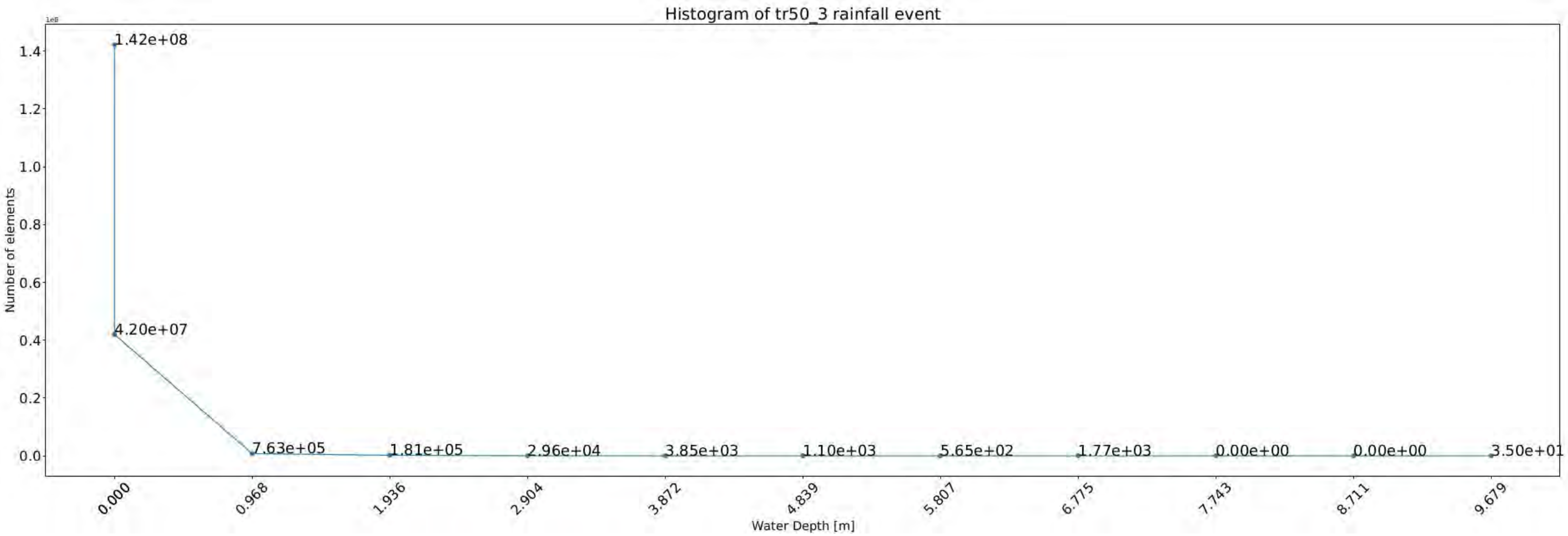
Res vs Pred uncertainty, tr50_3, 4500, patches



Res vs Pred uncertainty, tr50_3, 4500, patches



Histogram of tr50_3 rainfall event

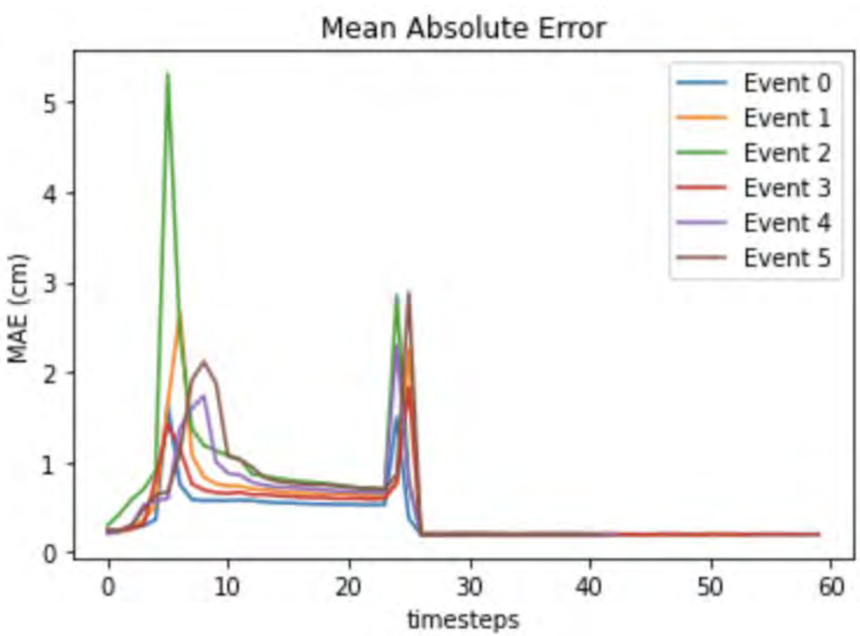


Update Summary

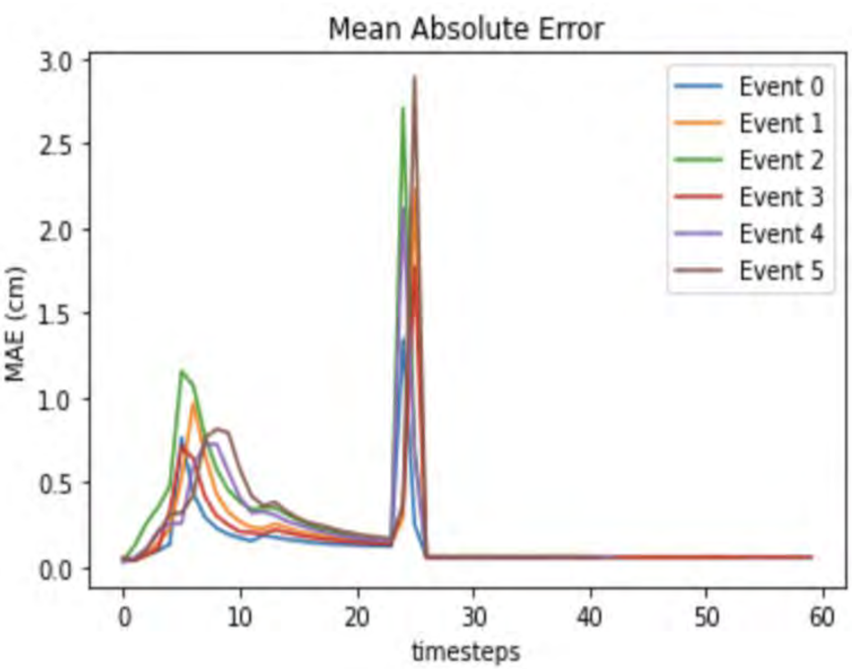
- Code is up-to-date and merged
- Experiments when taking *p%* or more data elements in patches is not generalizing well
    - Due to limited validation set size, and patches of p% data elements, model is not able to learn.
    - Either generate random sample for validation set with no minimum data element condition
    - Fix indices of validation set.

MAE(cm) for absolute value prediction vs. difference prediction

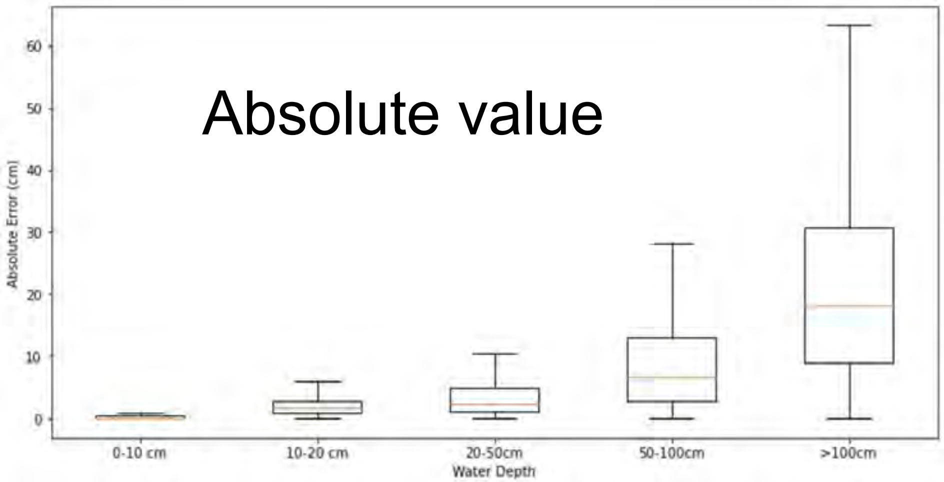
Absolute value

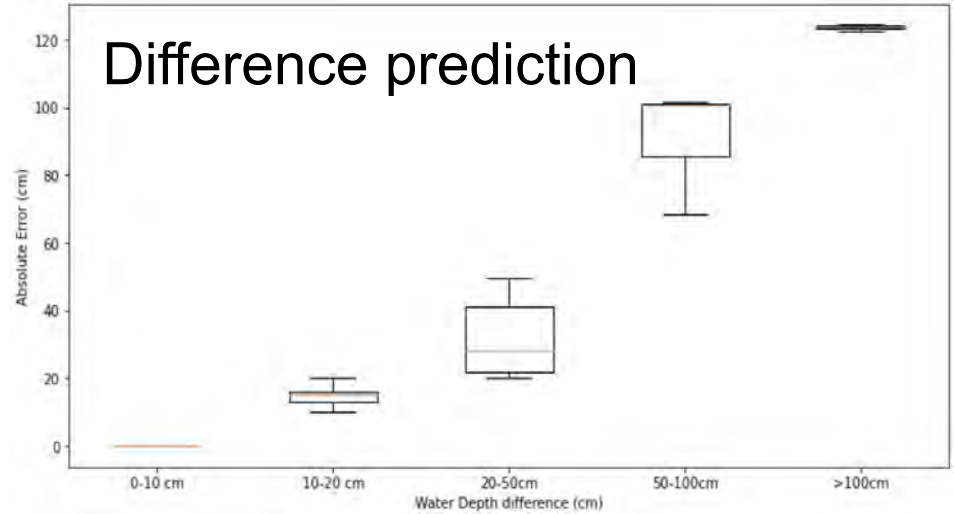


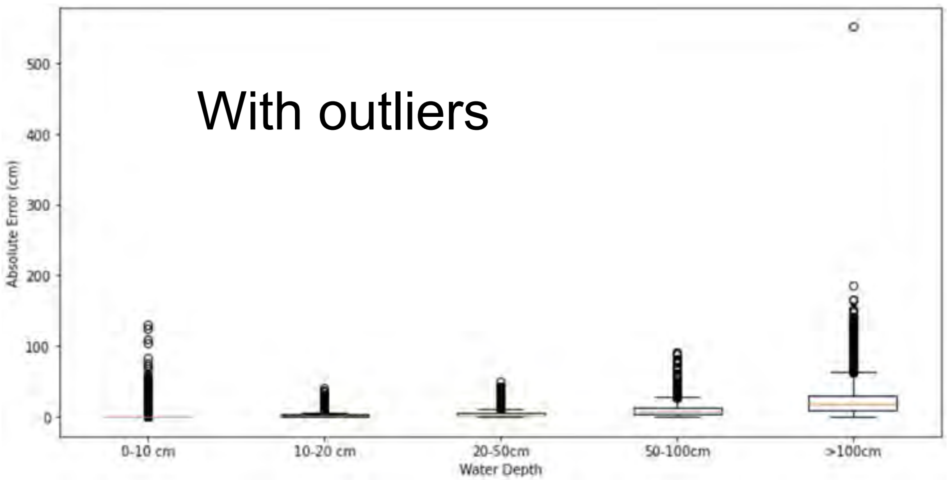
Difference prediction

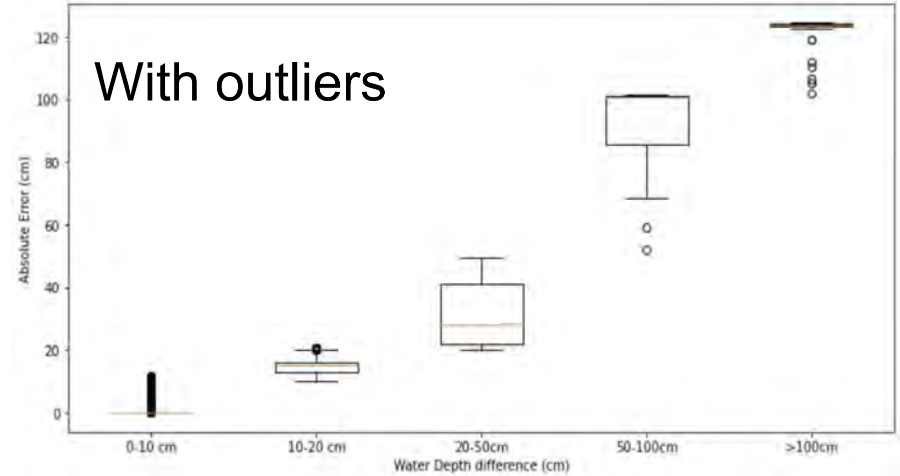


Boxplots(cm) for absolute value prediction vs. difference prediction

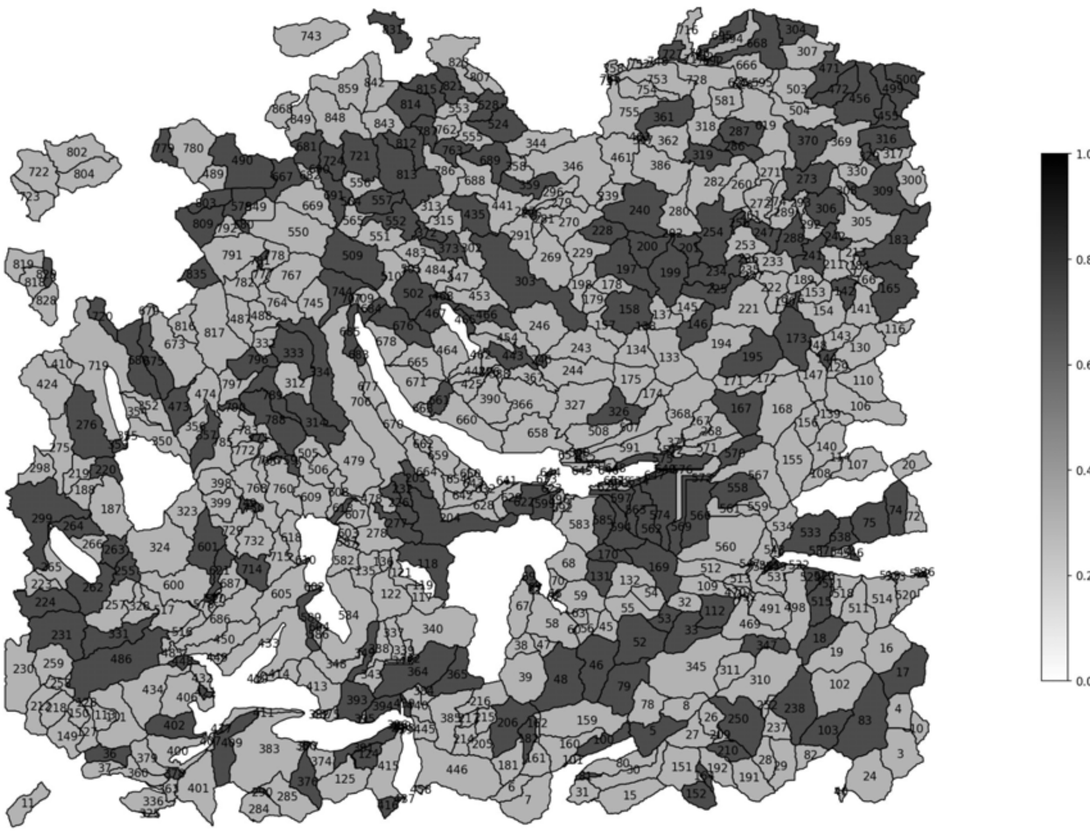








Further catchments to consider



Faster training - Joao's paper

Table 2. Average time performance of the prediction model

name	terrain image size (pixel)	terrain image pre-processing	[1] prediction time (no patch overlaps)	[1] prediction time (use mean value)	[1] prediction time (use median value)	[1] prediction time (use max value)	[1] simulation time	[2] training time
Luzern	3369 × 3110	1.898 s	0.678 s	2.693 s	14.749 s	2.556 s	2 h 20 min	5 h 25 min
Zurich	6175 × 6050	6.627 s	1.366 s	5.677 s	75.12 s	5.293 s	4 h 54 min	
Coimbra	1625 × 2603	0.636 s	0.242 s	0.965 s	5.048 s	0.902 s	2 h 18 min	

[1] The times are averaged and per rainfall event.

[2] For each catchment area, the amount of training data and training parameters were the same, and identical meta parameters were used; therefore, the average time is presented.

- "We tested our approach with only elevation and with all the features and found that introducing features makes training significantly faster."

Catchment Features

- Elevation
- Slope
- Aspect
- Curvature
- Mask

- The **slope** is defined as the magnitude of the gradient vector at each raster cell
- The **aspect** identifies the direction of the water flow at each raster cell and is the directional component of the gradient vector.
- The **curvature** is defined as the second derivative of the polynomial where two meaningful values can be calculated: the profile and plan curvature.