

Track 2 Hackathon ECO-AI

MinusCO2 Group

Reza Najafi Silab, Judongyang Zhou, Xinyee Tai, Yuanjing Zhao, Boyang Chen

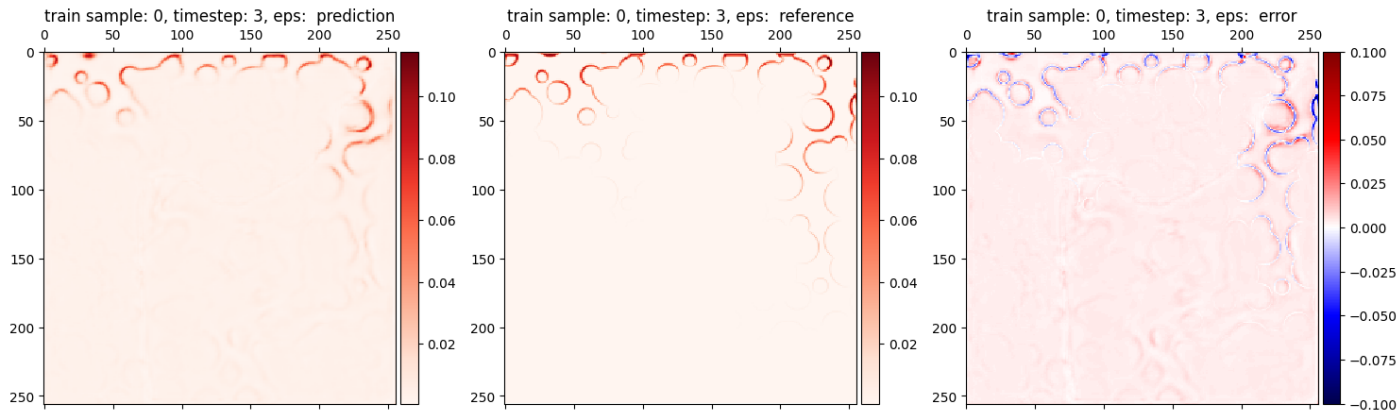
15/03/2024

Content

- Problem Description
- Solutions
- Results & Conclusion
- What we have learned?

Problem Description

Training

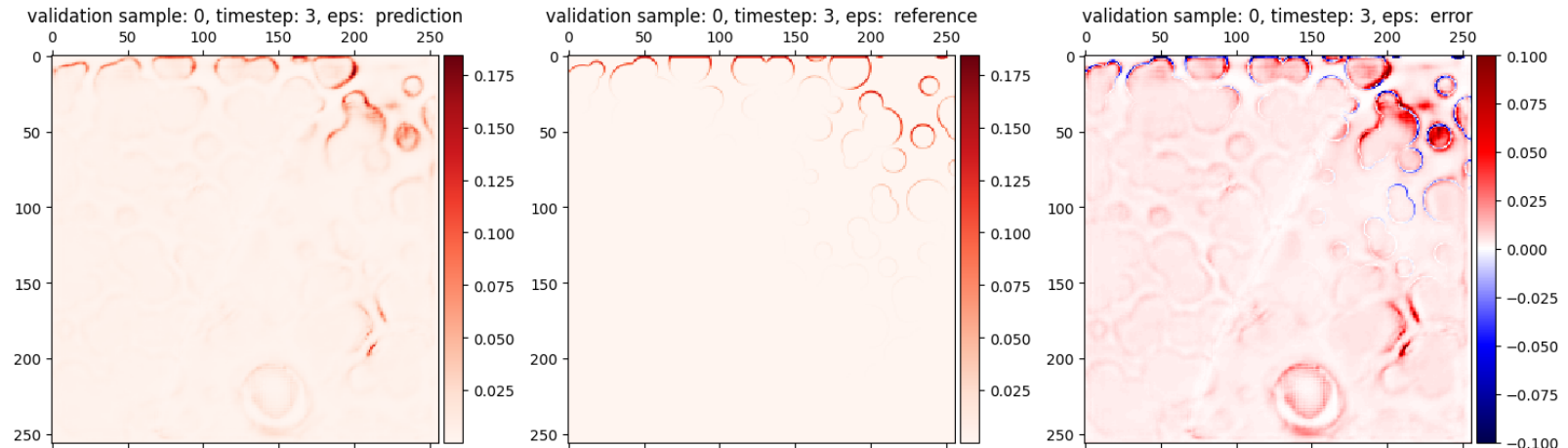


Overfitting

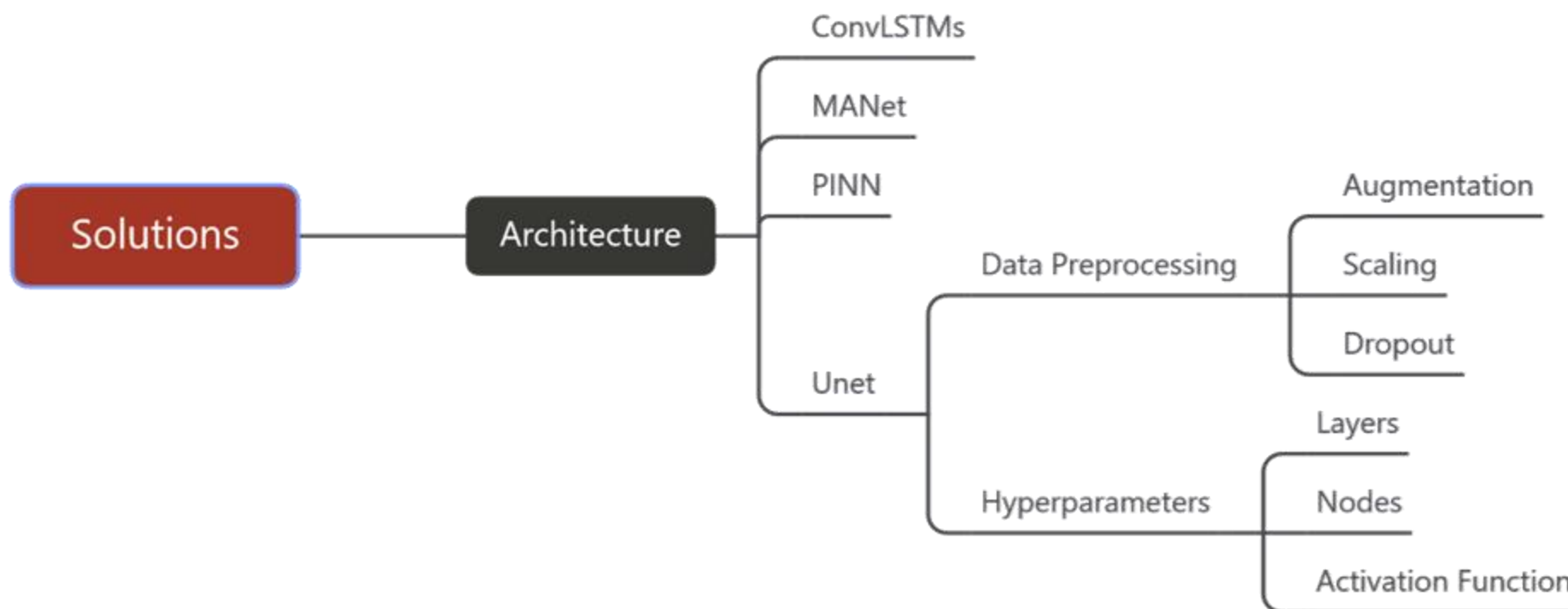


Validataing

Original U-net:
train_loss: $2.1e-5$;
validation_loss: $1.9e-4$

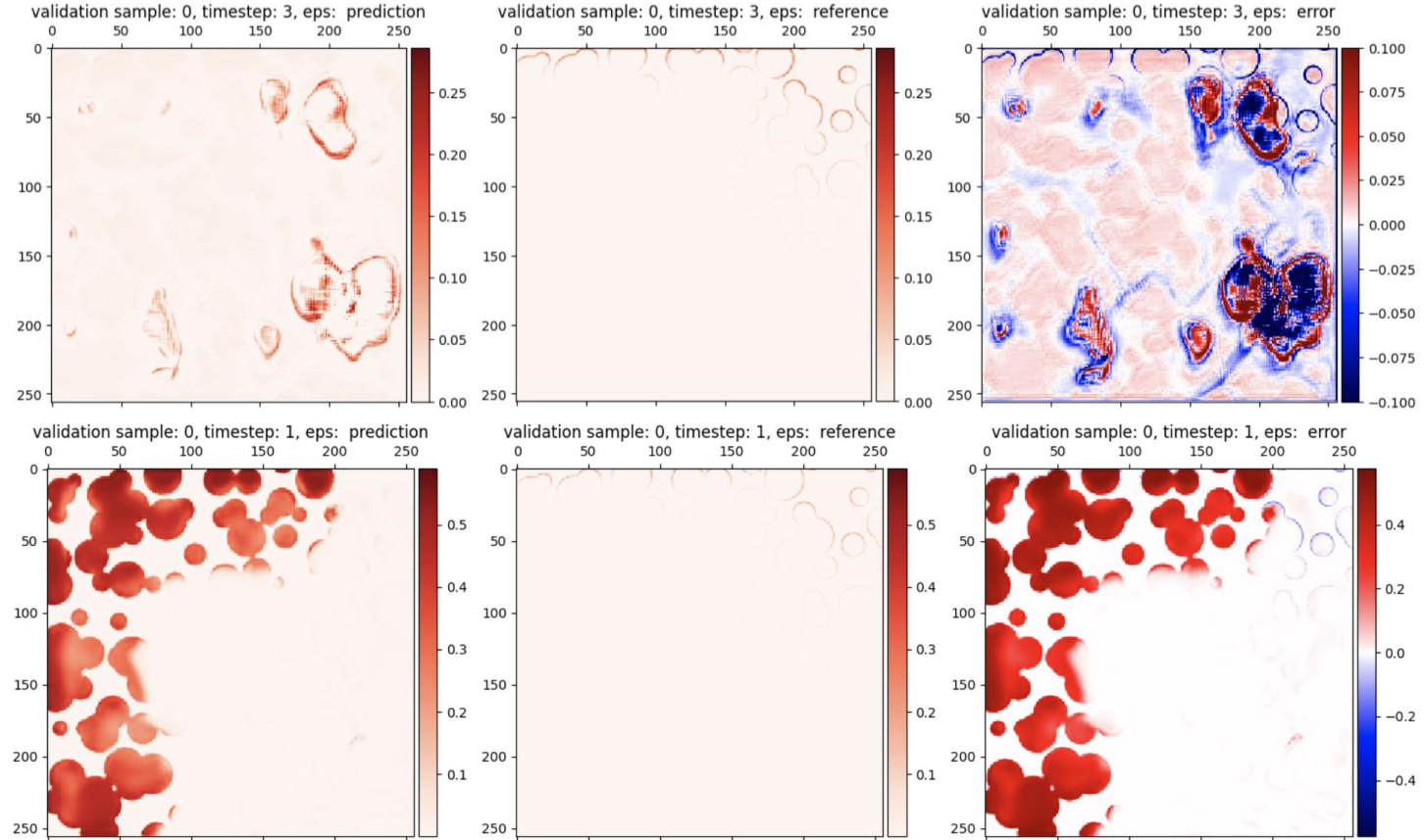


Solutions



Result & Conclusion

- Slightly better:
 - activation function
- Even worse:
 - MANet
 - augmentation
 - scaling



What we have learned?

Dongyang

- Over ambitious:
 - attempt to try ConVLSTM & PINN even if I had no experience before
 - from EASY to HARD next time

OutOfMemoryError: CUDA out of memory. Tried to allocate 256.00 MiB. GPU 0 has a total capacity of 19.71 GiB of which 65.56 MiB is free. Including non-PyTorch memory, this process has 19.09 GiB memory in use. Of the allocated memory 18.91 GiB is allocated by PyTorch, and 882.00 KiB is reserved by PyTorch but unallocated. If reserved but unallocated memory is large try setting max_split_size_mb to avoid fragmentation. See documentation for Memory Management and `PYTORCH_CUDA_ALLOC_CONF`

- data augmentation:
 - Flip, rotation, transpose the data makes my neural network worse (because they changed the domain?)

Reza

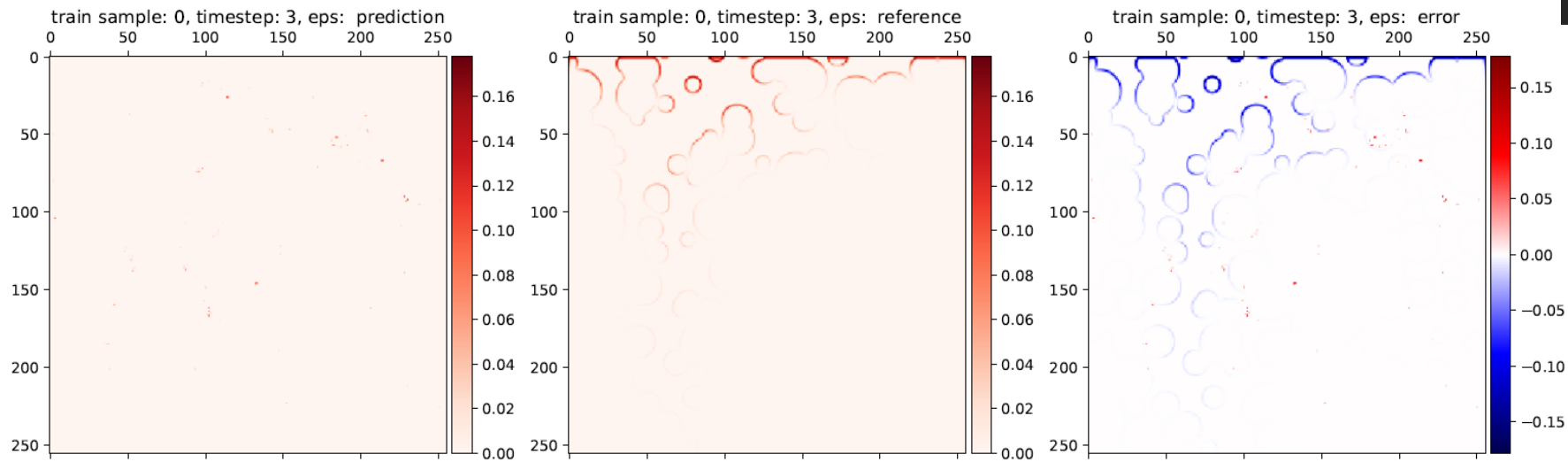
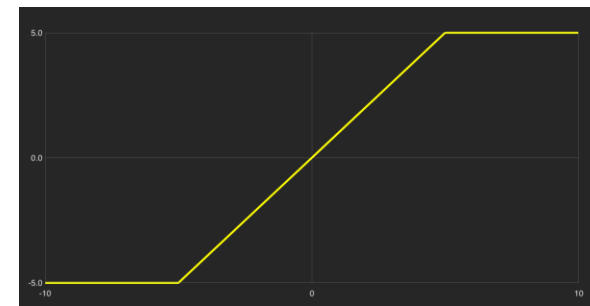
- PINNs
 - Why they were not suitable for our problem?
 - Checking the physics validity in our domain
- A couple of new architectures
 - RNN
 - ConvLSTM

Yuanjing

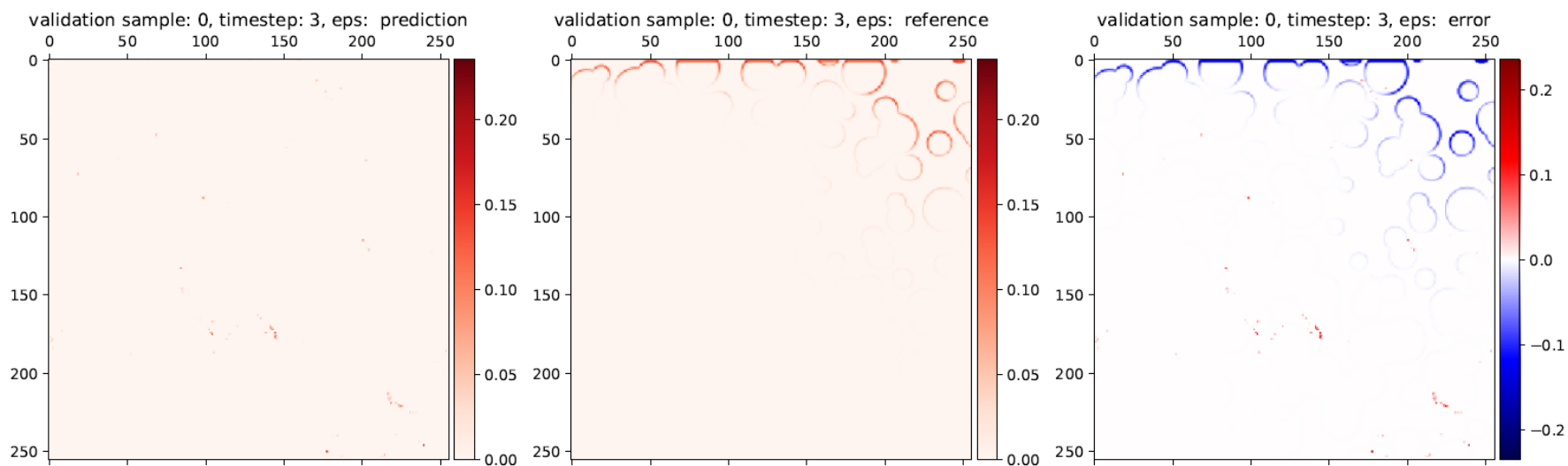
- ML(Multilayer perceptron)
- DNN
- convLSTM
- Optuna
- Overcome overfitting problem
- Start with sth small
- Hyperparameters

Xin yee

function: Clamps
Activation

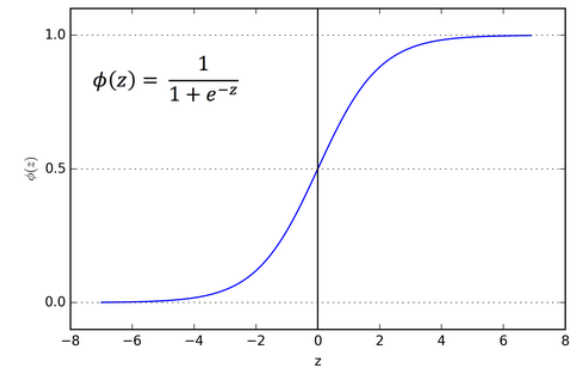
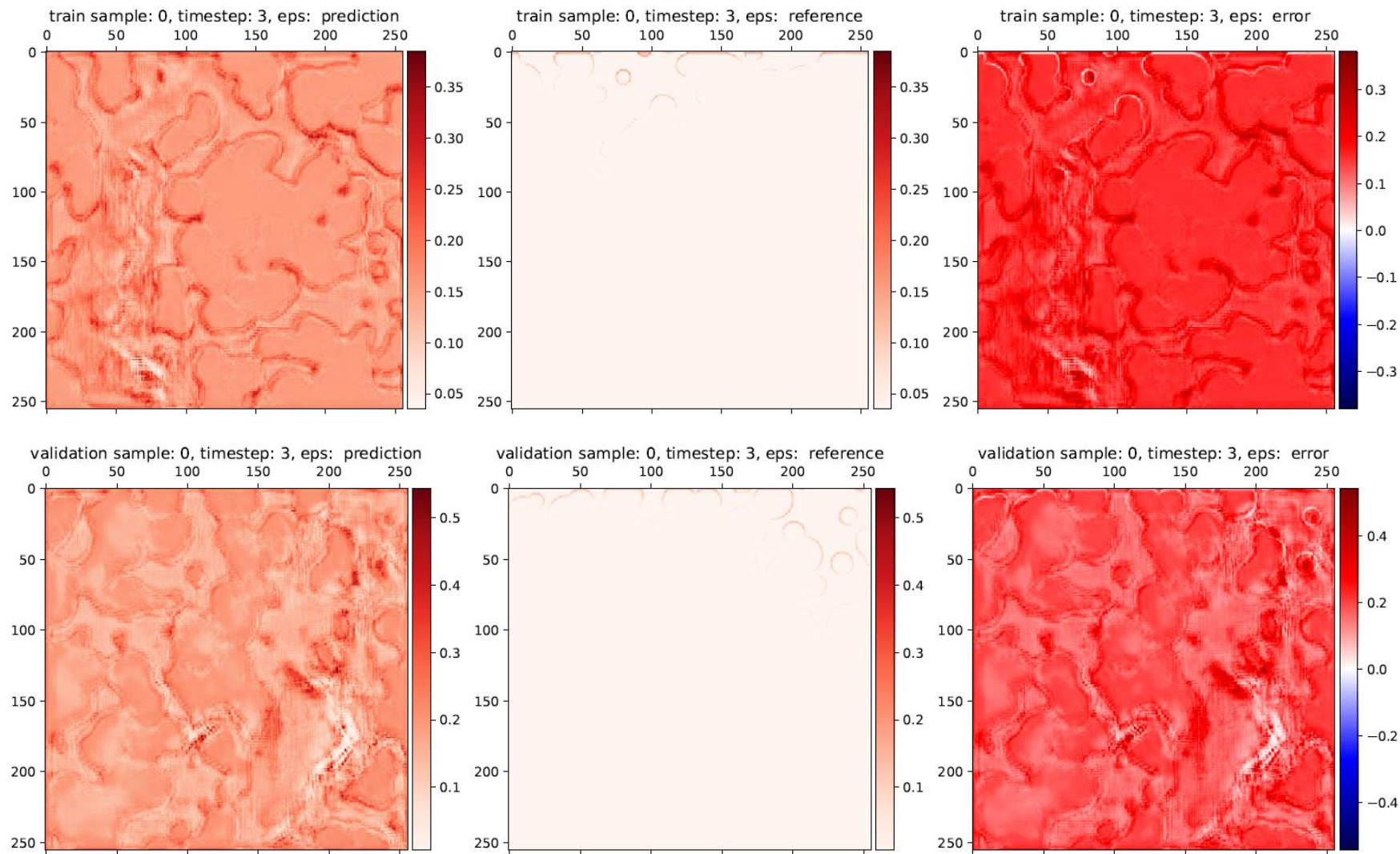


Training loss: 0.00011



Validation loss: 0.0026

Activation function: Sigmoid



Training loss: 0.0269

Validation loss: 0.0341

Thanks!