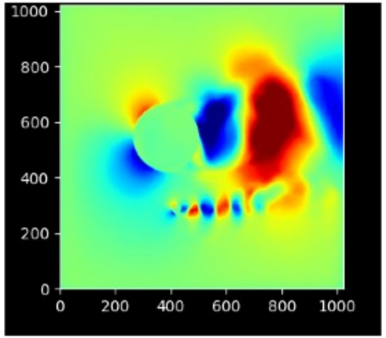
Fast Prediction Methods for Fluid Simulation Results Using Deep Neural Networks

Takashi Shimokawabe

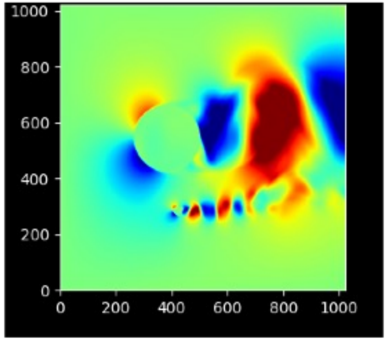
Information Technology Center, The University of Tokyo

Our group aim: Accelerating MD/CFD by DL

■Computational Fluid Dynamics

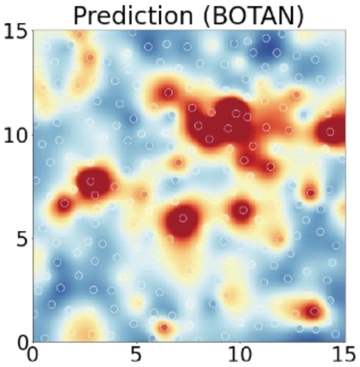


CNN prediction

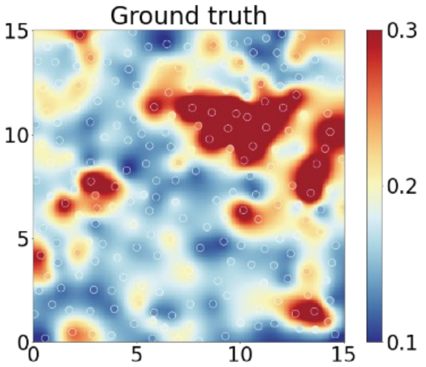


CFD Simulations

■Molecular Dynamics



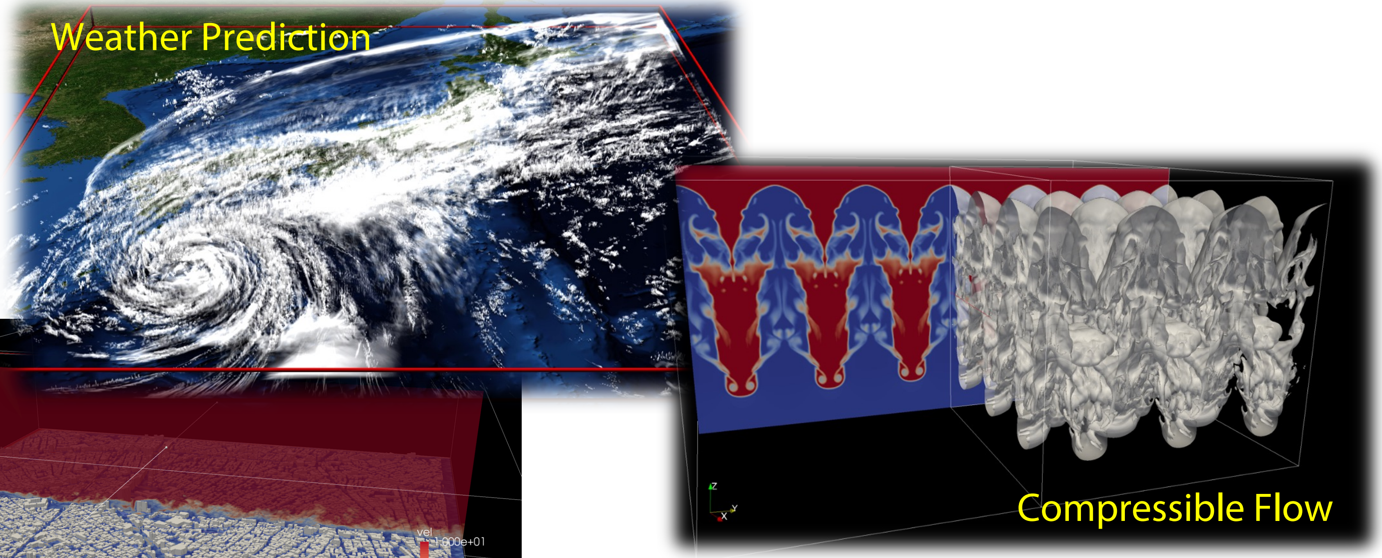
GNN prediction

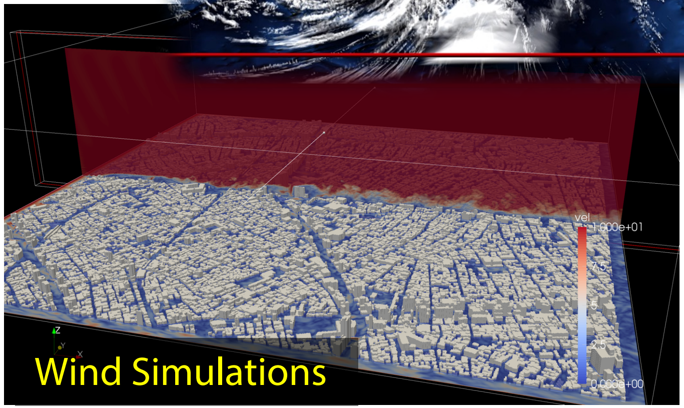


MD Simulations

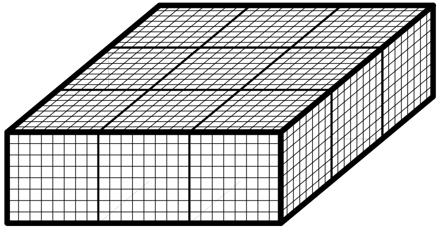
16:35 - 16:50	**Hayato Shiba (Online) (University of Hyogo)** Deep learning of simulated glassy dynamics

Target: Computational Fluid Dynamics

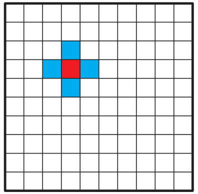




Computational cost of computational fluid dynamics is relatively high.



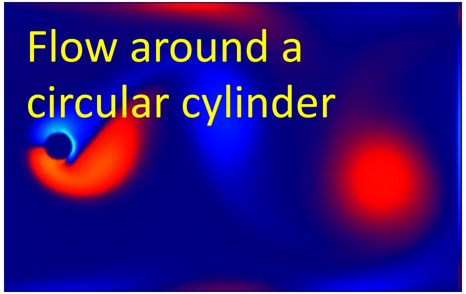
Computational domain



Stencil computations

Fast prediction of CFD simulation results by DNN

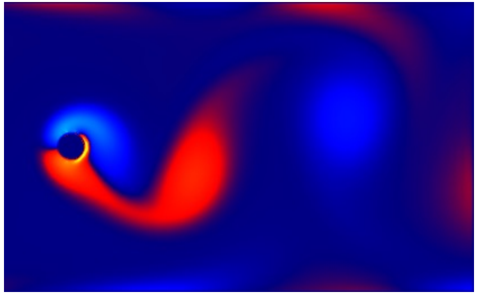
Dataset



CFD simulation
(Lattice Boltzmann
methods)

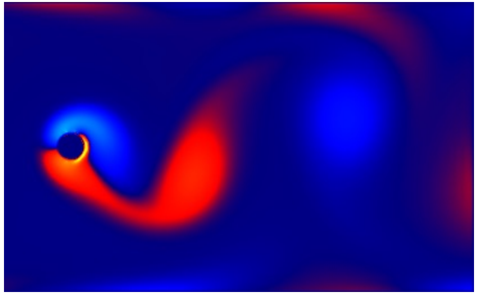
$$f_i(x + c_i \Delta t, t + \Delta t) = f_i(x, t) + \Omega_i(x, t)$$

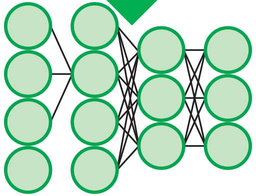
$$\Omega_i(x, t) = -\frac{1}{\tau}(f_i(x, t) - f_i^{eq}(x, t))$$



Training

Prediction





Prediction of flow

Convolutional neural networks (CNNs) to
predict simulation results

CNNs may become “faster simulator”

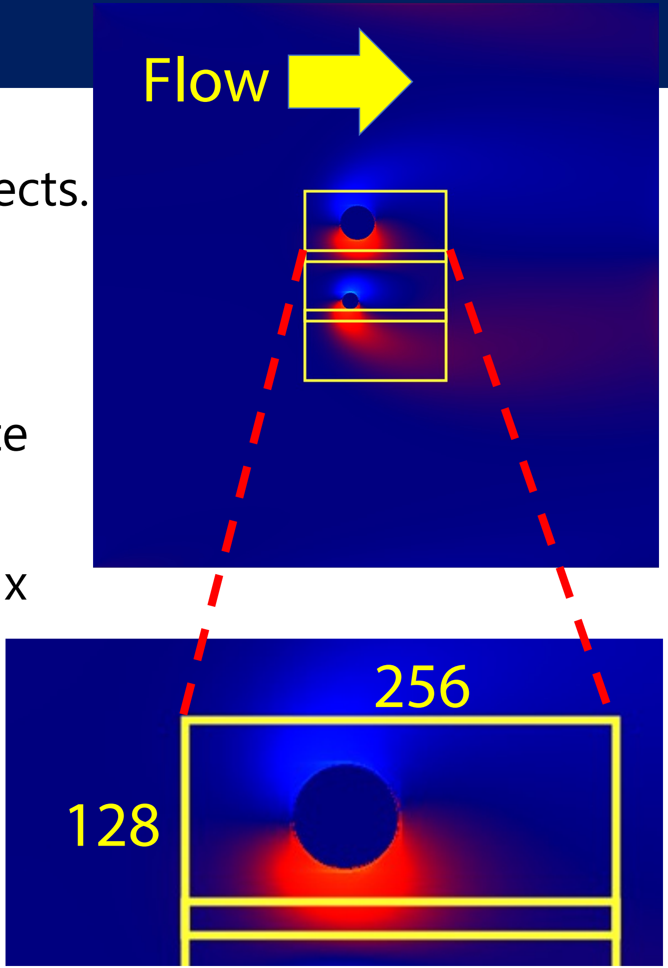
Topic1: Steady flow

■Steady flow

- The fluid flows along the x axis around objects.

■LBM (Lattice Boltzmann method) simulation results

- D2Q9 model (9 variables is used for discrete velocity)
- Re = 20,
- Region size: 256 x 128 (clipped from 1024 x 1024)
- 6 types of object shapes:
  - polygons (number of angles: 3-7)
  - cylinders.



■Input data: 256 x 128 (clipped)

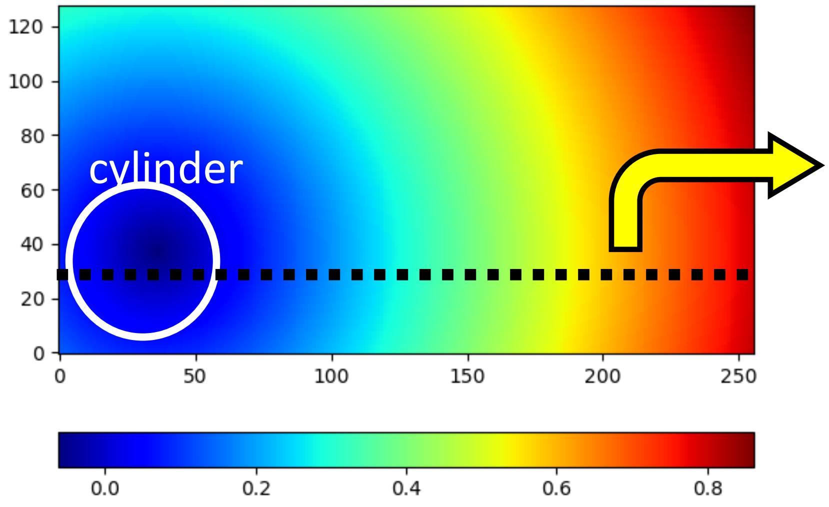
- Training: 14515
- Validation: 1613

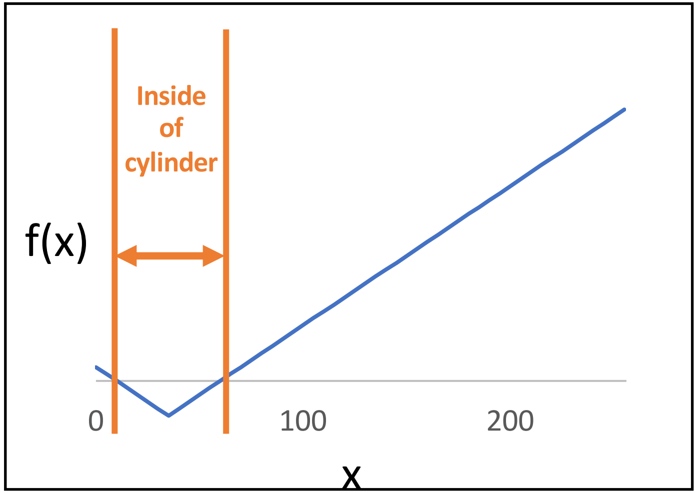
Singed distance function (SDF)

■ **SDF represents**

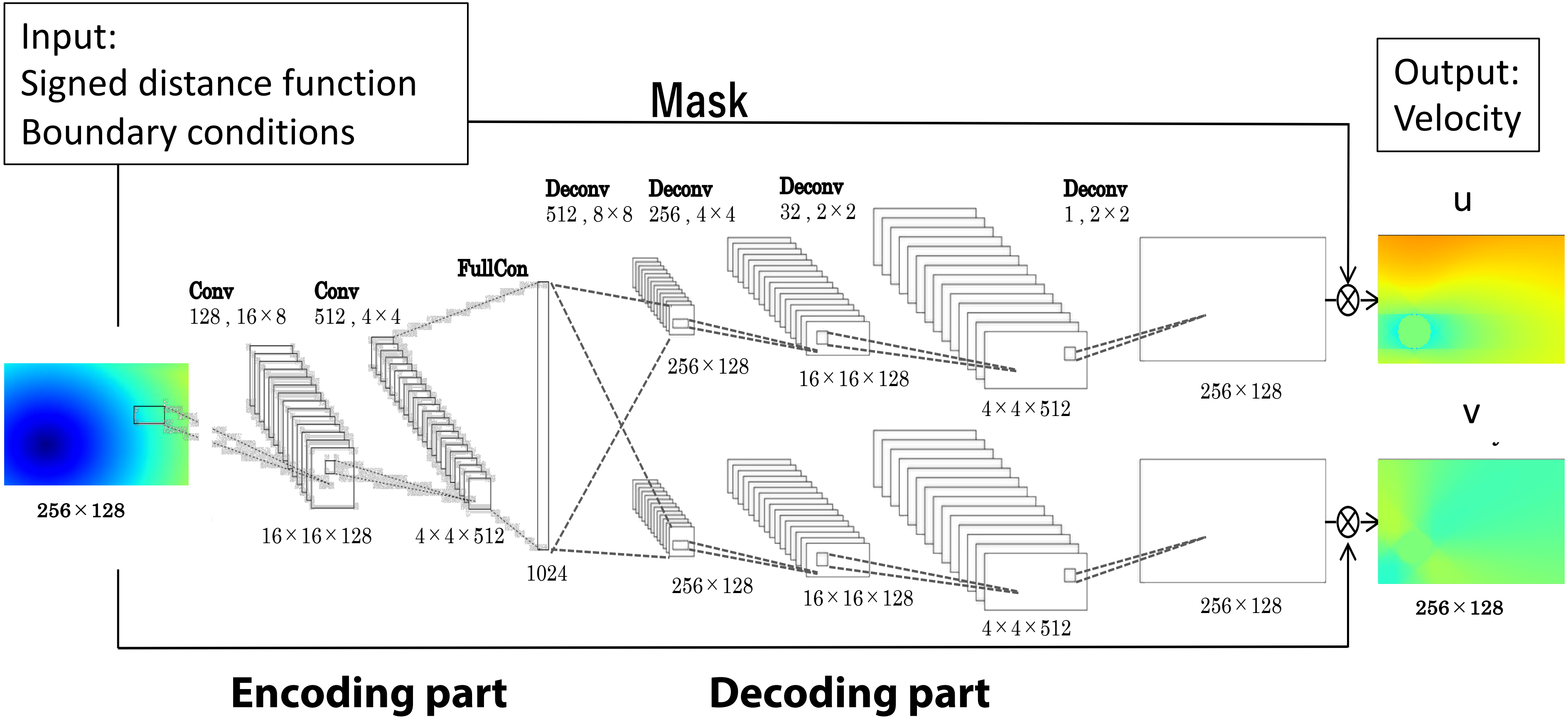
- the surface of the object as zero.
- the outside of the object as a positive distance.
- the inside of the object as a negative distance.

■ **A universal representation for different geometry shapes and works efficiently with neural networks**



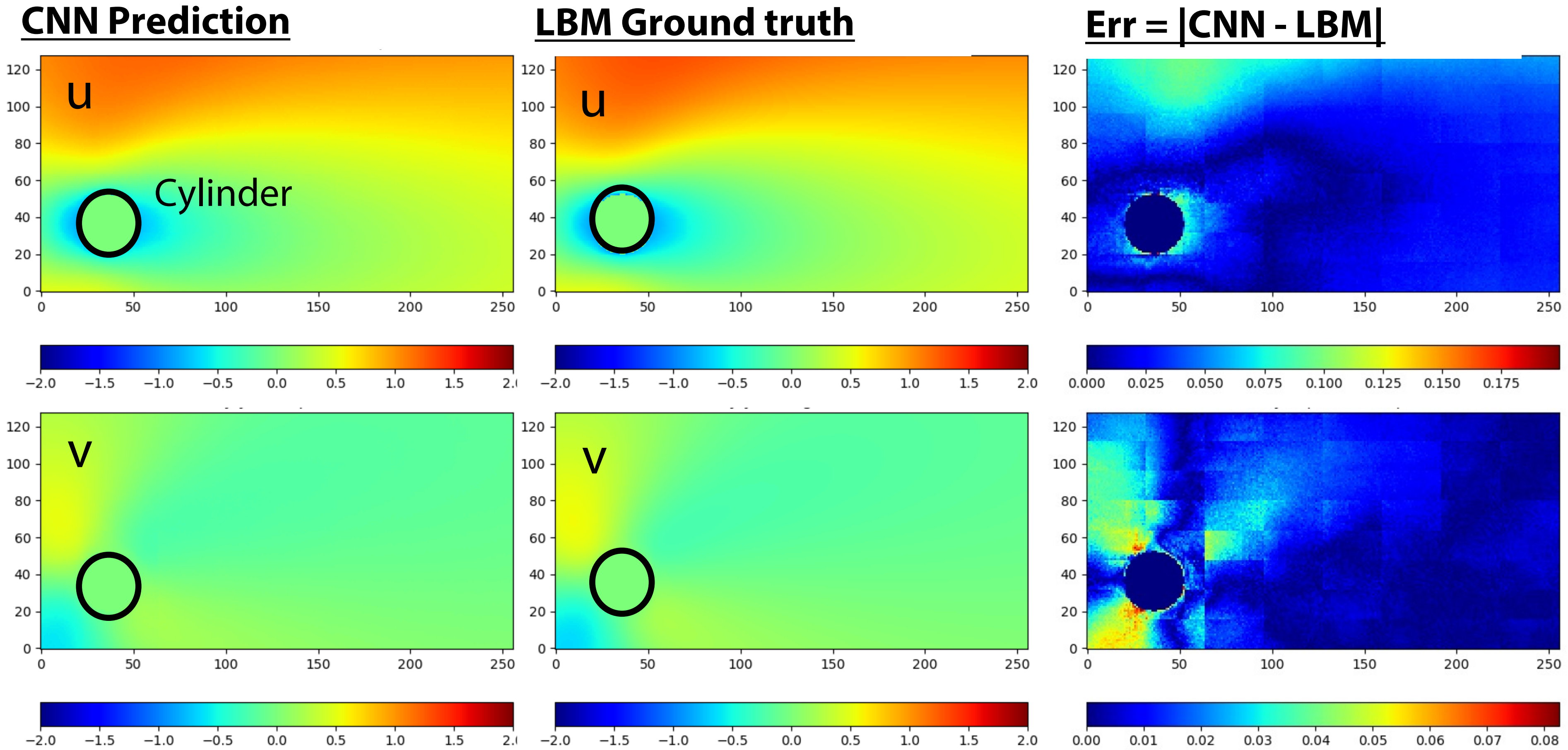


Network Architecture and Training



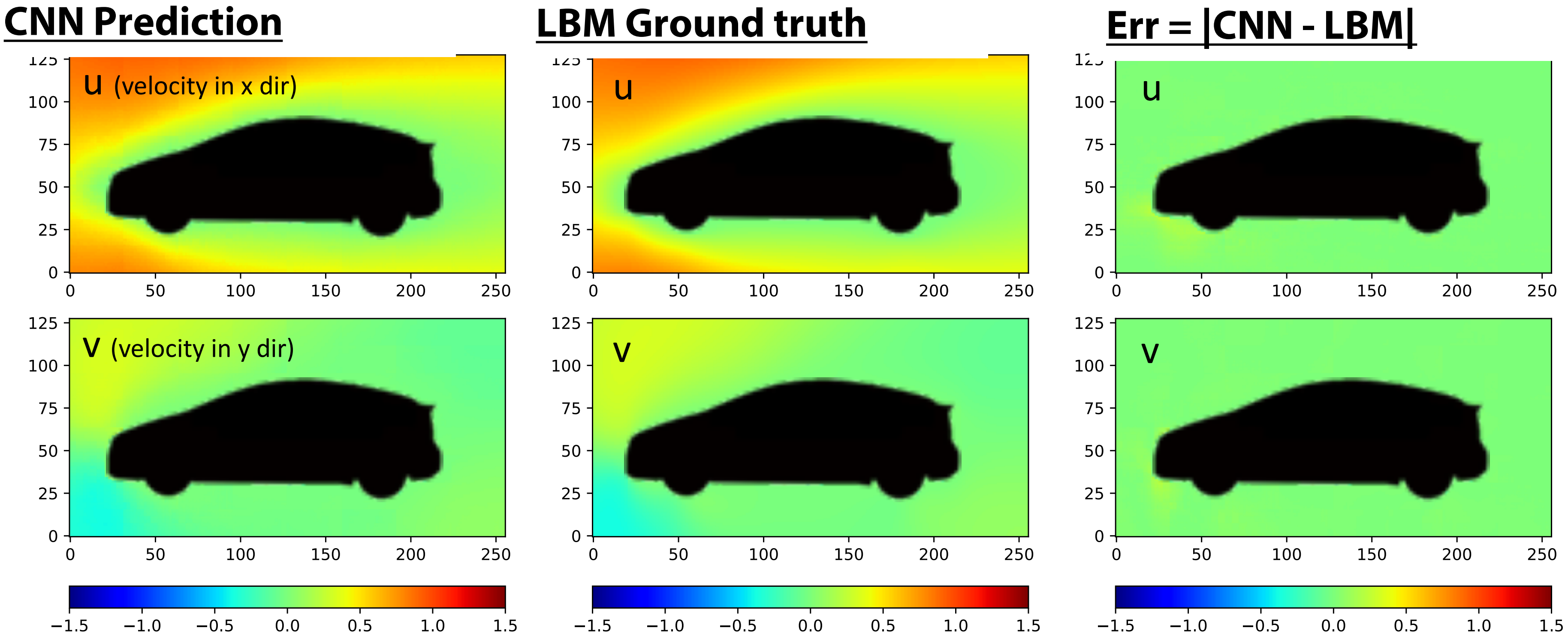
A modified version of the network architecture proposed in Guo et al.
"Convolutional Neural Networks for Steady Flow Approximation", 2016

Prediction results for single domain



Loss: 7.3 x 10^{-5}

Prediction results for a complex shape

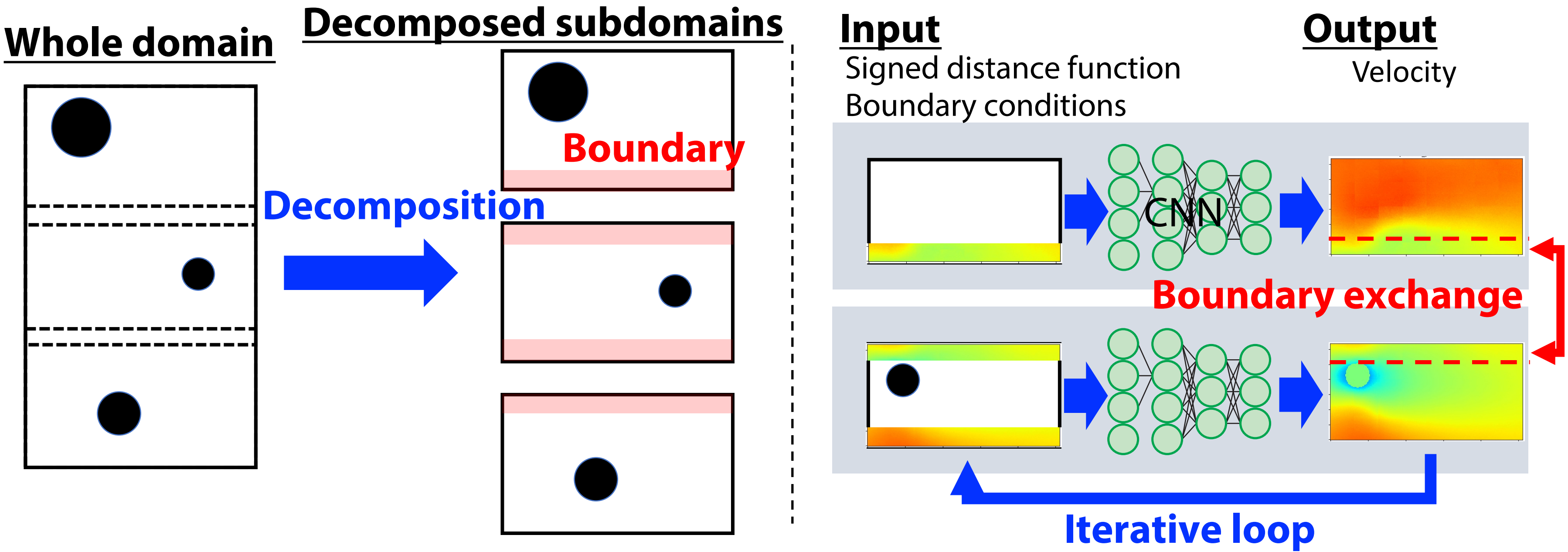


- **Computation time…** LBM (82,000steps) → 41.1 sec 、 CNN prediction → 0.6 sec

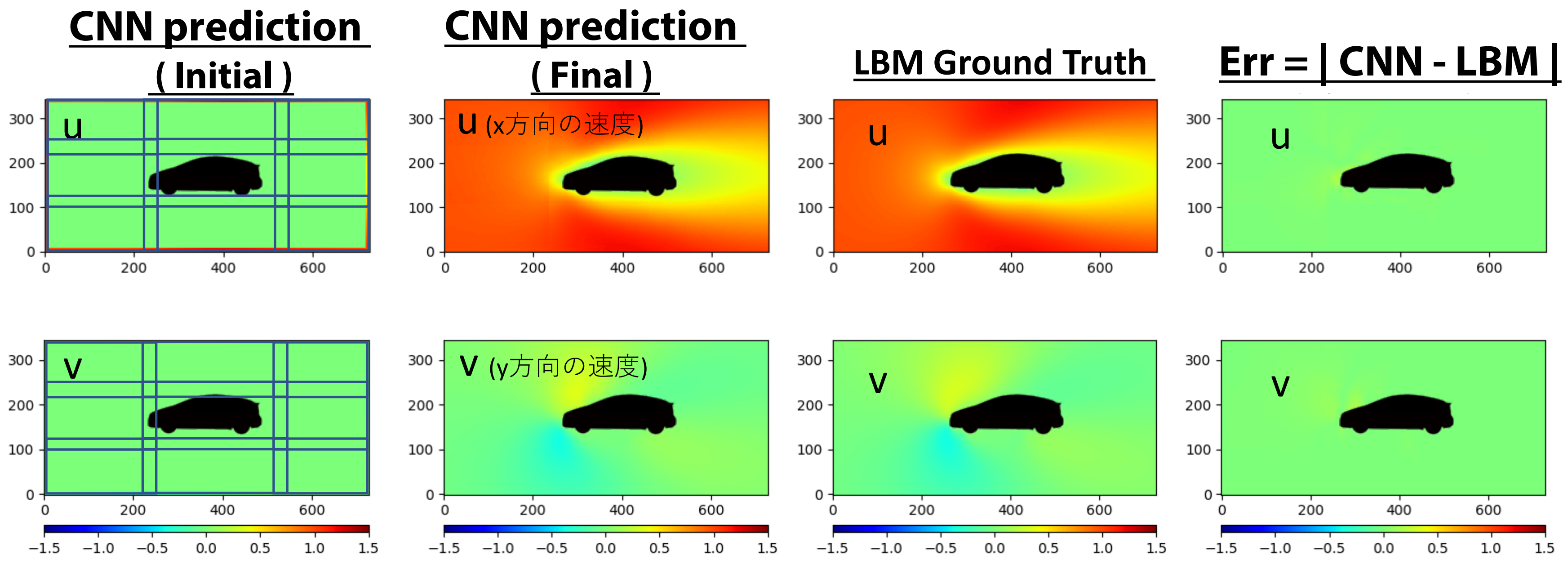
⇒ CNN prediction has achieved high accuracy with significant reduction in calculation time.

Prediction by CNN with boundary exchange

- The network model trained for a single domain is applied to the decomposed subdomains to predict the simulation results in each subdomain.
- In order to maintain consistency between values in the subdomains, boundary exchange between neighbor subdomains is performed.
- CNN and boundary exchange are performed iteratively until values converge.
- This method has no limitation for device (GPU) capacity.



Predicted results using CNN with boundary exchange



Domain size ：748 x 364 (9 decomposed subdomains)

Mean error ：3.89％

Comp. time ：3.82 s

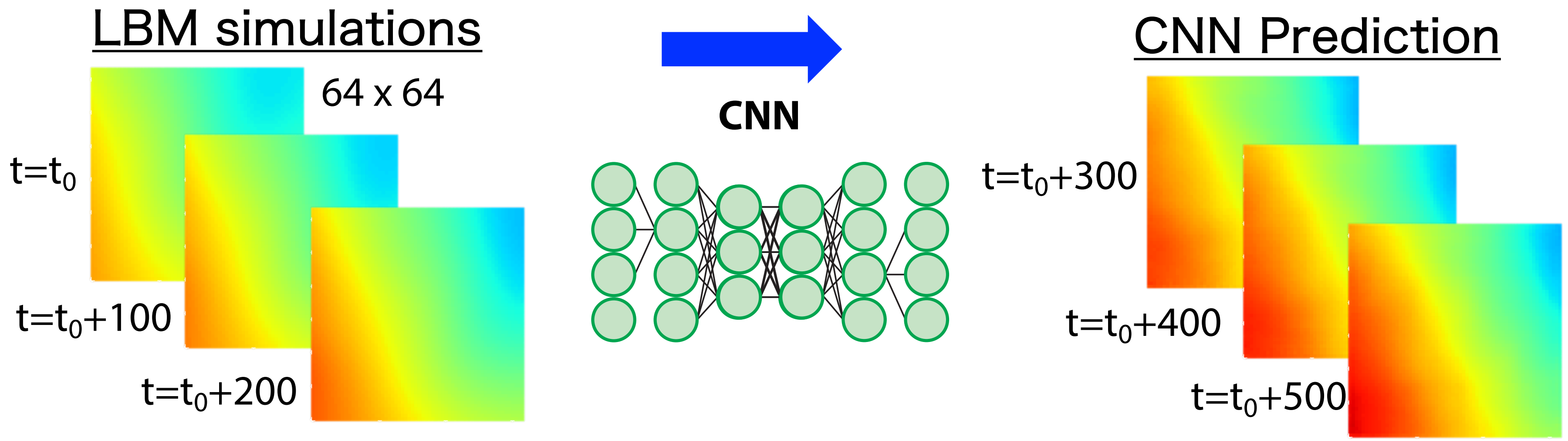
Topic2: Time dependent flow

■3D CNN prediction of time dependent flows

- 2 dimensions in space and 1 dimension in time : x, y, t → 3D CNN

■Input and output

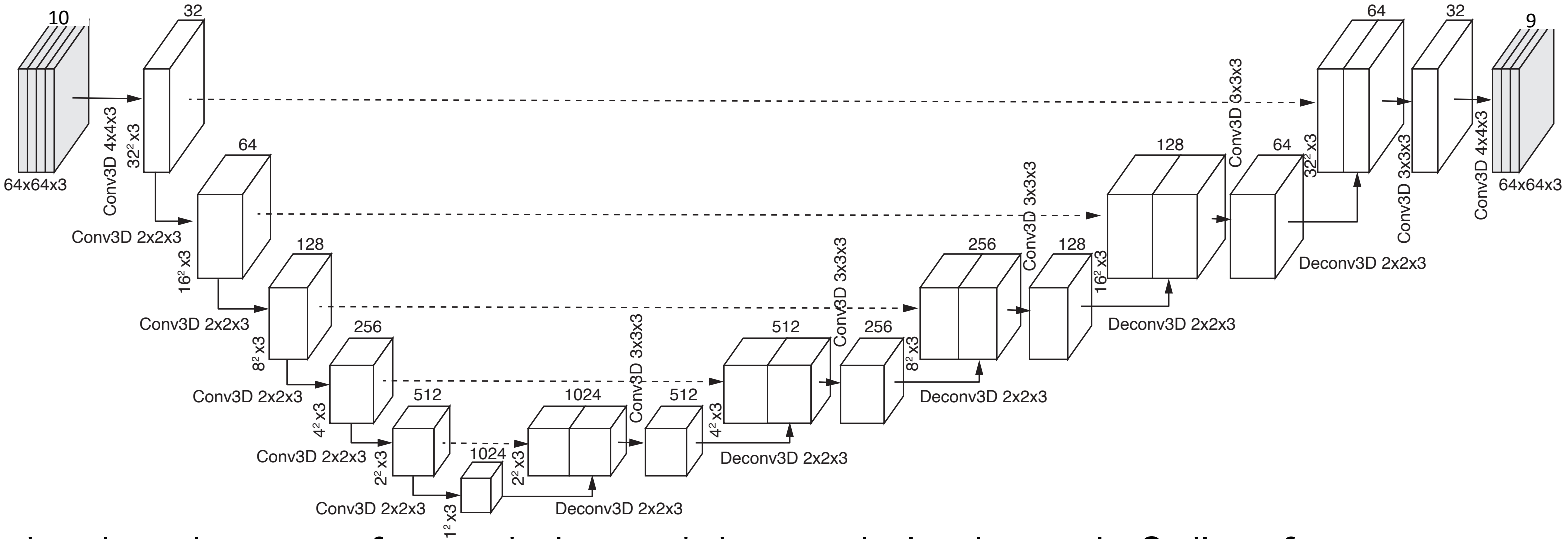
- Input: 3 frames (t0, ..., t0+200) at 100 step intervals in the LBM simulation
- Output: Predicts the next 3 frames (t0+300, ..., t0+500)



Network structure to predict time dependent flow

Input:
 discrete velocity distribution function
 Singed distance function (geometry shapes)

Output:
 discrete velocity distribution function



- Encoder-decoder type of convolution and deconvolution layers in 3 dim of x, y, t
- Activation function: tanh
- Skip connections (used in U-net, etc.)
- Normalization of input/output data
- Loss function based on collision process in LBM

Lattice Boltzmann Method

A Discrete-velocity Boltzmann equation

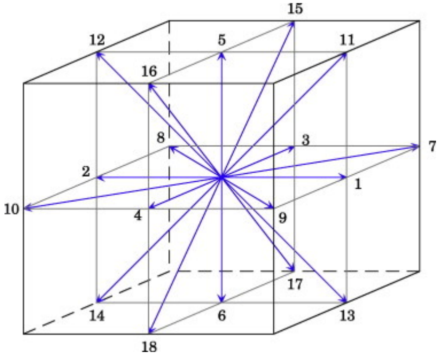
$$\underbrace{f_i(x + c_i\Delta t, t + \Delta t) = f_i(x, t)}_{\text{streaming}} - \frac{1}{\tau}\underbrace{(f_i(x, t) - f_i^{eq}(x, t))}_{\text{collision}} + \underbrace{F_i}_{\text{forcing term}}$$

Streaming step

f_i : discrete velocity distribution function

c_i : discrete velocity

D3Q19
discrete velocity

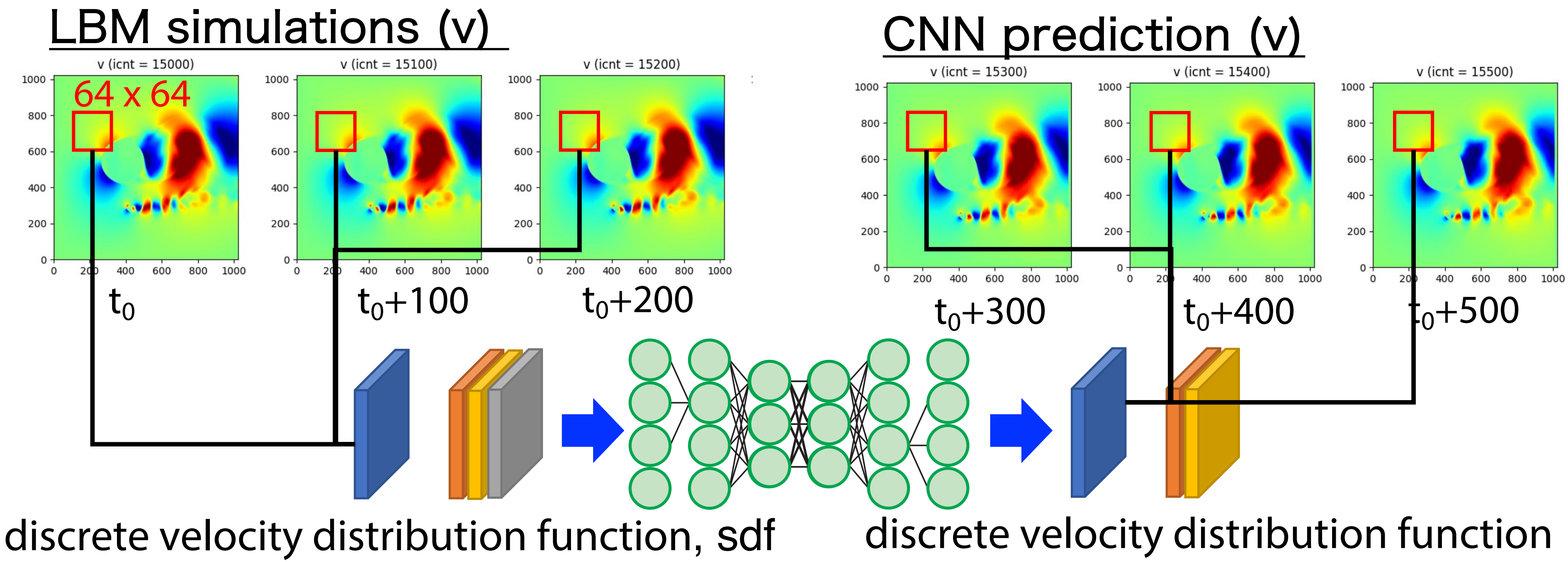


velocity	time-integration (n → ＊)
$(0,0,0)$	$f_0^*(x,y,z) = f_0^n(x,y,z)$
$(c,0,0)$	$f_1^*(x,y,z) = f_1^n(x - c\Delta t, y, z)$
$(-c,0,0)$	$f_2^*(x,y,z) = f_2^n(x + c\Delta t, y, z)$
$(0,c,0)$	$f_3^*(x,y,z) = f_3^n(x, y - c\Delta t, z)$
⋅	⋅
$(0,0,c)$	$f_5^*(x,y,z) = f_5^n(x, y, z - c\Delta t)$
⋅	⋅
$(c,c,0)$	$f_7^*(x,y,z) = f_7^n(x - c\Delta t, y - c\Delta t, z)$
⋅	⋅
⋅	⋅

copy

Predicts large simulation results

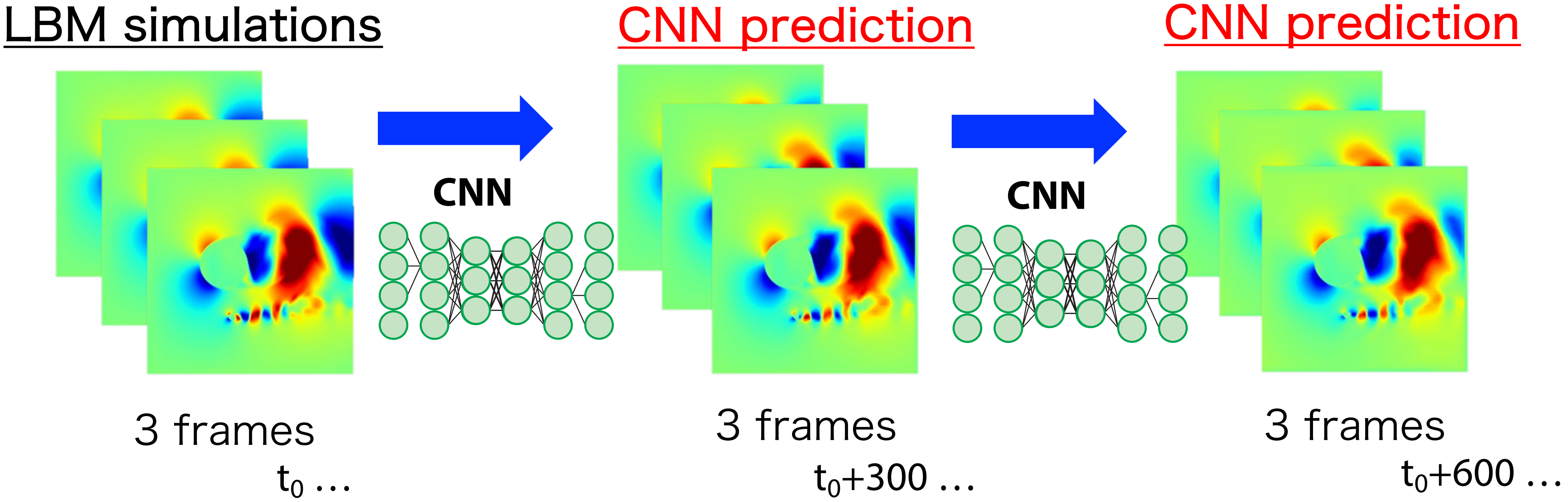
- Predictions are needed for regions beyond the training size
- Apply inference patchily over the entire computational domain
- Apply inference in an overlapping regions



Prediction with multiple applies of CNN inference

■Prediction of long time evolution by CNN

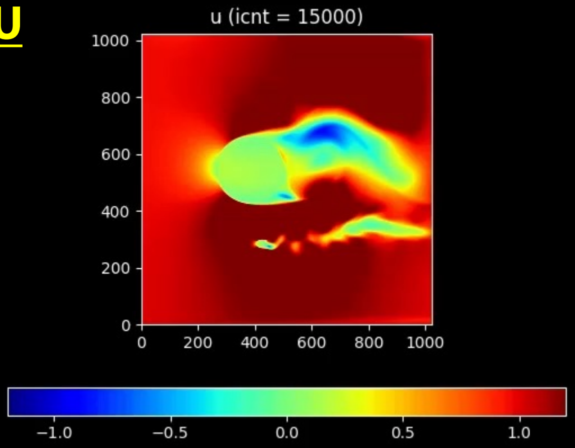
- Predicts further frames ahead by using the CNN prediction result as the next input



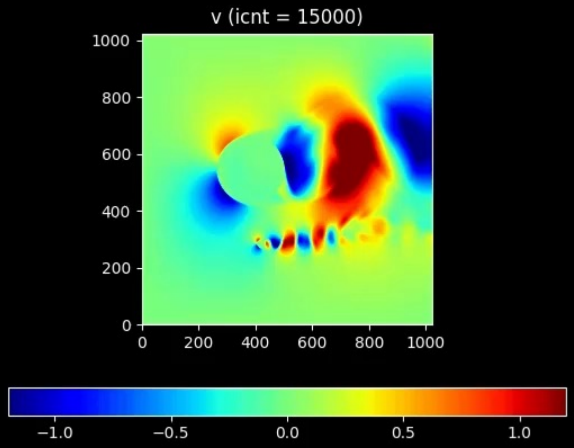
Prediction Example 1

LBM Ground truth

U



V

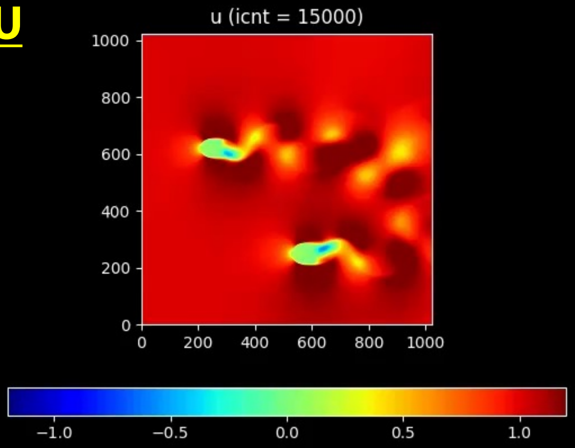


CNN Prediction

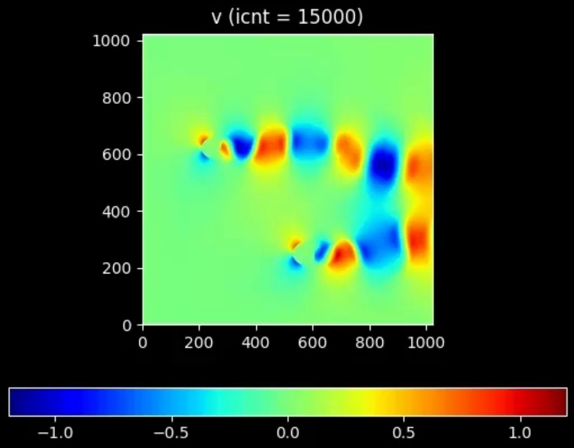
Prediction Example2

LBM Ground truth

U



V



CNN Prediction

Conclusion

- **We have developed deep learning fast prediction methods for steady/unsteady flow.**
- **Steady flow**
  - Predicting simulation results on large domain using CNN with boundary exchange.
  - The proposed method has no limitation for device (GPU) capacity.
- **Time dependent flow (unsteady flow)**
  - Predicting simulation flow using 3D CNN.
  - By applying the CNN multiple times, long time predictions can be made.
  - Applied patchily, predictions can be made for regions beyond the size of the training data.