jh210049-MDH

Supercomputer resources:
FX1000 @ Nagoya Univ, Tsubame3.0 @Tokyo Tech, BDEC @ Univ. Tokyo

Developping data driven analysis methods for extreme scale numerical simulation

Representative:	**<u>Y. Asahi (JAEA)</u>**	Code development
Deputy Representative:	S. Maeyama (Nagoya Univ.)	Plasma turbulence
Deputy Representative:	J. Bigot (MdS, France)	Scalable data analysis
Collaborating researcher:	X. Garbet (CEA, France)	Global plasma turbulence
Collaborating researcher:	V. Grandgirard (CEA, France)	Large scale simulation
Collaborating researcher:	K. Fuji (Kyoto Univ.)	Machine learning
Collaborating researcher:	T. Shimokawabe (Univ Tokyo,)	Deep learning
Collaborating researcher:	T.-H. Watanabe (Nagoya Univ.)	Local plasma turbulence
Collaborating researcher:	Y. Idomura (JAEA)	Large scale simulation
Collaborating researcher:	N. Onodera (JAEA)	Large scale simulation
Collaborating researcher:	Y. Hasegawa (JAEA)	Large scale simulation
Collaborating researcher:	T. Aoki (Tokyo Tech.)	Optimization on GPU
Collaborating researcher:	T. Katagiri (Nagoya Univ.)	Optimization on CPU

JHPCN 13th symposium, Shinagawa, Japan **Date: 9/July/2021**

Objectives

Scalable Data analysis

- Large scale data analysis based on Dask
- Analyzing the time series of 5D distribution function [1] ★
- In-situ machine learning to avoid saving the huge data

AI 4 CFD

- Convolutional neural network to predict the multi-resolution steady flow data [2]
- Data-driven modeling of Subgrid scale dynamics

Data driven analyses for Exascale simulations

[1] Y. Asahi et al., Phys. Plasmas, 2021
[2] Y. Asahi et al., to be submitted

★ Completed in JH200065 "Modernizing and accelerating fusion plasma turbulence codes targeting exa-scale systems"

Objectives

Scalable Data analysis

- Large scale data analysis based on Dask
- Analyzing the time series of 5D distribution function [1] ★
- In-situ machine learning to avoid saving the huge data

AI 4 CFD

- Convolutional neural network to predict the multi-resolution steady flow data [2]
- Data-driven modeling of Subgrid scale dynamics

Data driven analyses for Exascale simulations

[1] Y. Asahi et al., Phys. Plasmas, 2021
[2] Y. Asahi et al., to be submitted

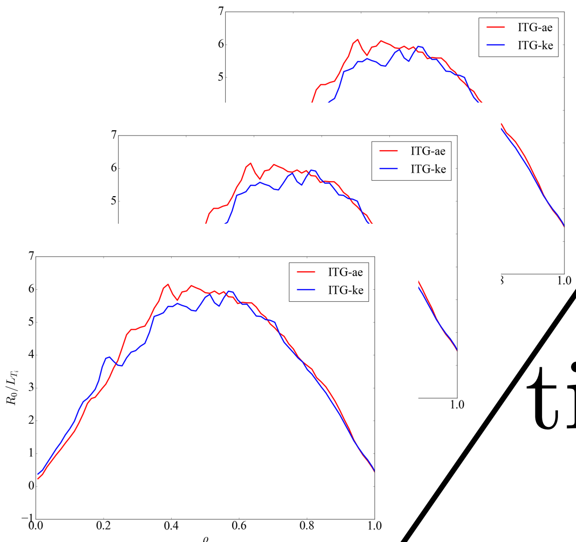
★ Completed in JH200065 "Modernizing and accelerating fusion plasma turbulence codes targeting exa-scale systems"

Analyzing 5D gyrokinetic simulation data

Conventional study | **This work**

1D time series

Structures of radial profile



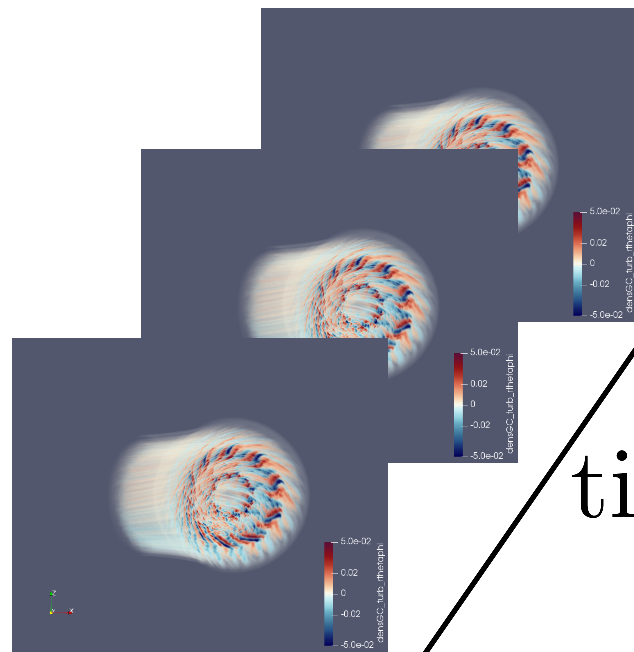
time

ρ

~kB

3D time series

Structures of Fluid moments δn



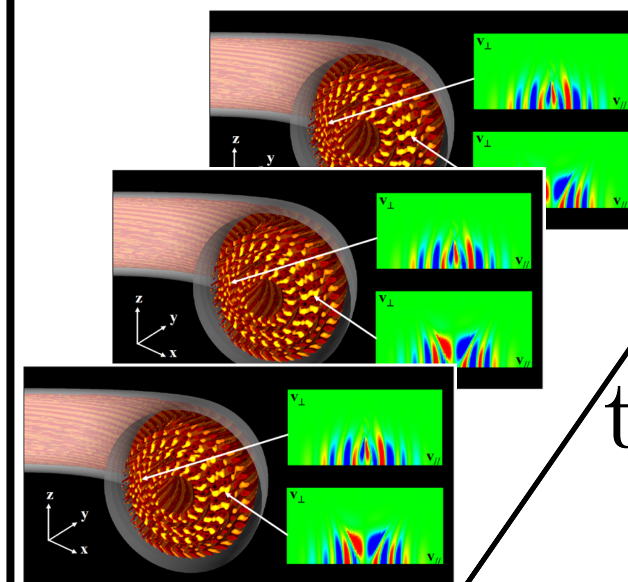
time

~10MB

5D time series

Phase structure

δf



time

~10GB

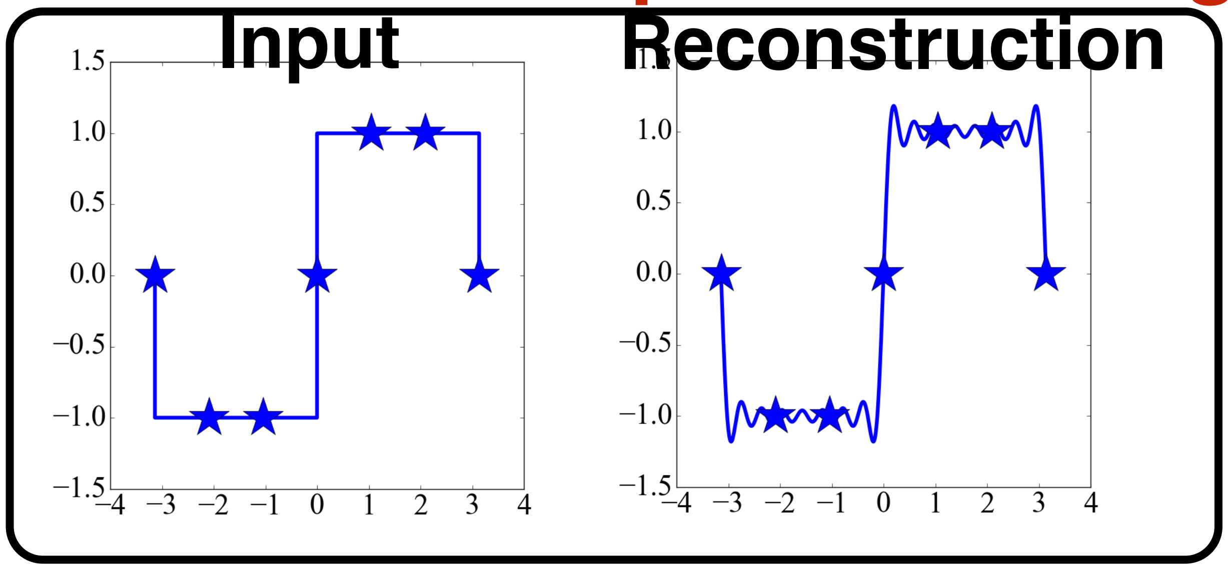
High dimensional + huge data

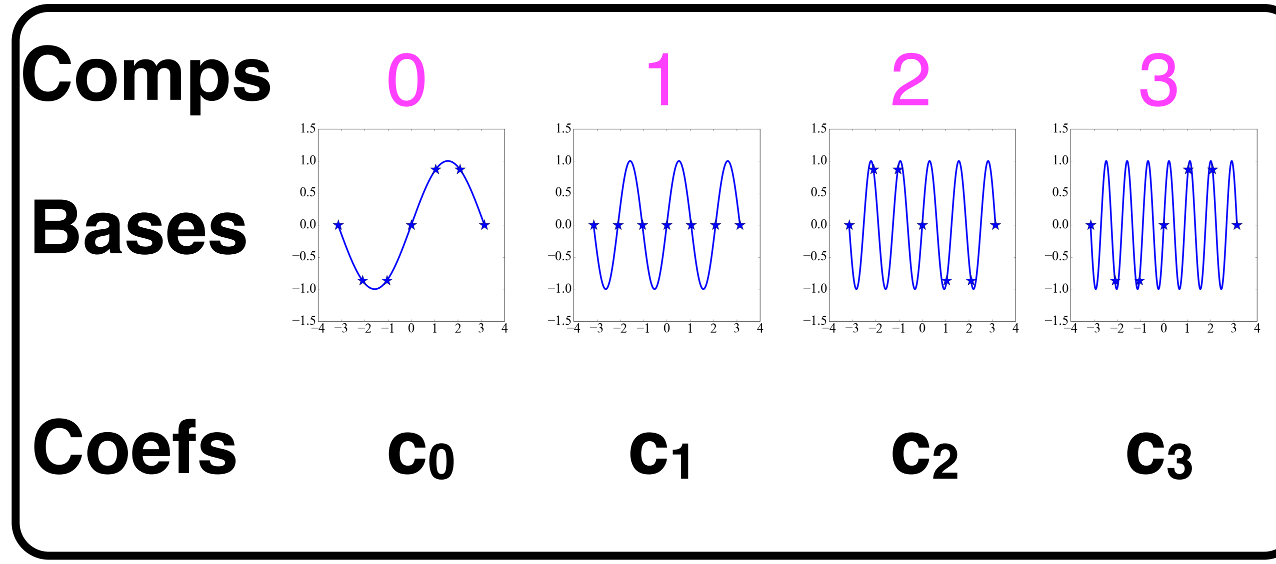
Conventional Study: 3D structures (like convective cells), 1D structures (stair case, stiffness in temperature gradient)

This work: Extracting phase space structure from the time series of 5D distribution function (pattern formation in phase space)

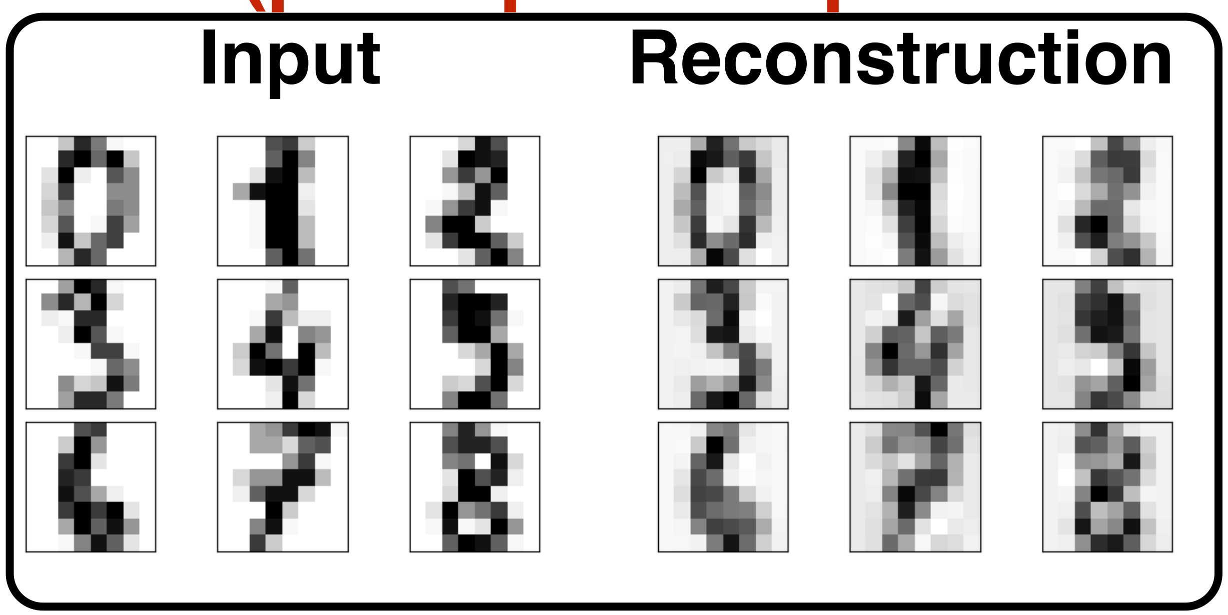
PCA and Fourier Transform

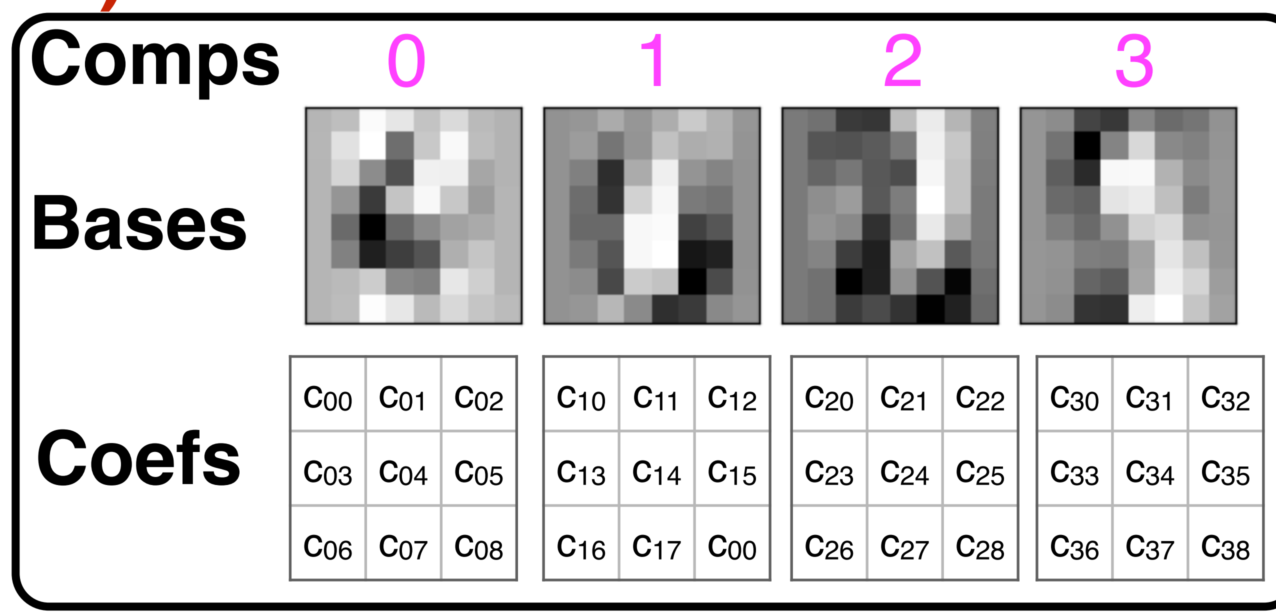
Fourier decomposition on signals





PCA (principal component analysis) on hand-written numbers





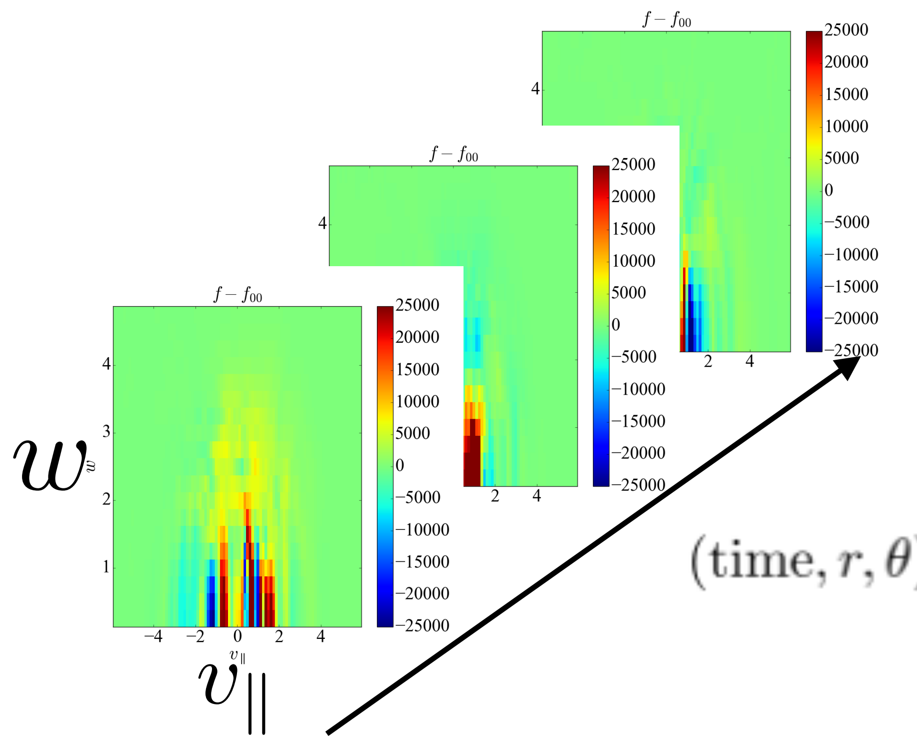
	Input	Bases	Coefficients	Reconstruction
FFT on signals	$N_{\text{signals}} \times (\text{scalar})$	$(N_{\text{signals}}, N_{\text{basis}}) \times (\text{scalar})$	$N_{\text{basis}} \times (\text{scalar})$	$\text{Sig}\,(m) = \sum_n C_n I_n\,(m)$
PCA on numbers	$N_{\text{numbers}} \times (\text{width, height})$	$N_{\text{basis}} \times (\text{width, height})$	$(N_{\text{numbers}}, N_{\text{basis}}) \times (\text{scalar})$	$\text{Img}\,(m,x,y) = \sum_n C_{n,m} I_n\,(x,y)$

Dimensionality reduction keeping important features in the data

PCA on Tera byte scale data with Dask+Xarray

Input $(\mathrm{time}, r, \theta) \times (\varphi, v_{\|}, w)$

A Samples Features



DASK

xarray

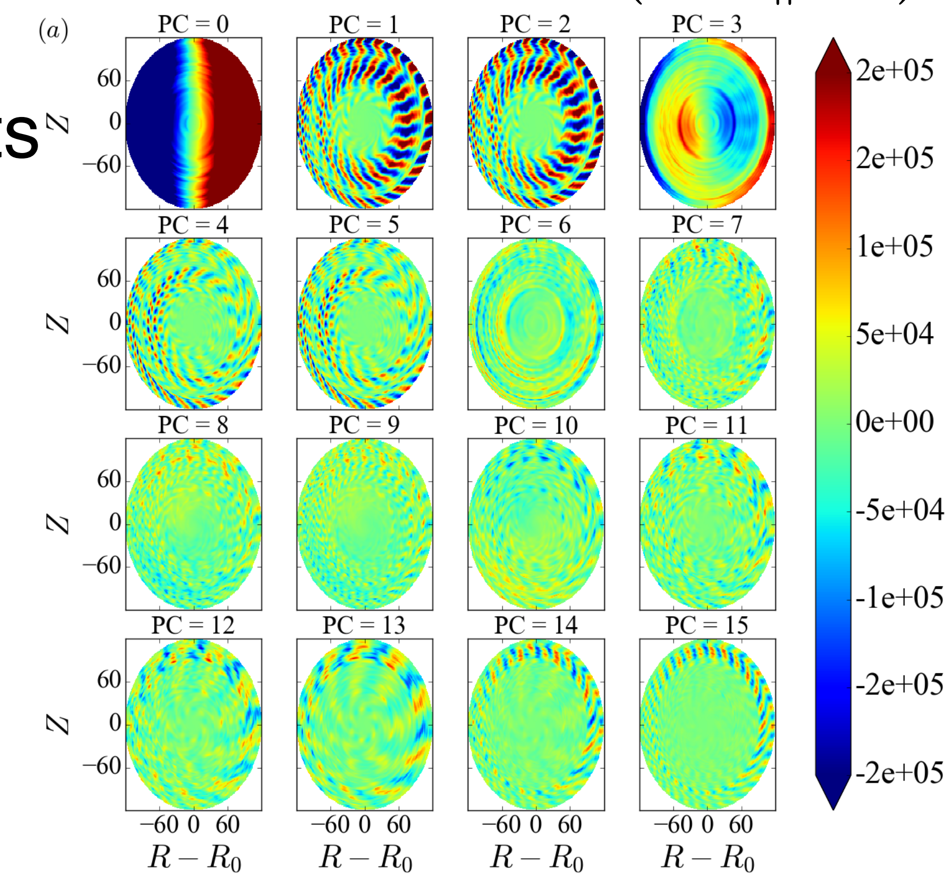
PCA

$A \sim U\Sigma V^T$

Output $(\mathrm{components}) \times (\varphi, v_{\|}, w)$

$U\Sigma$

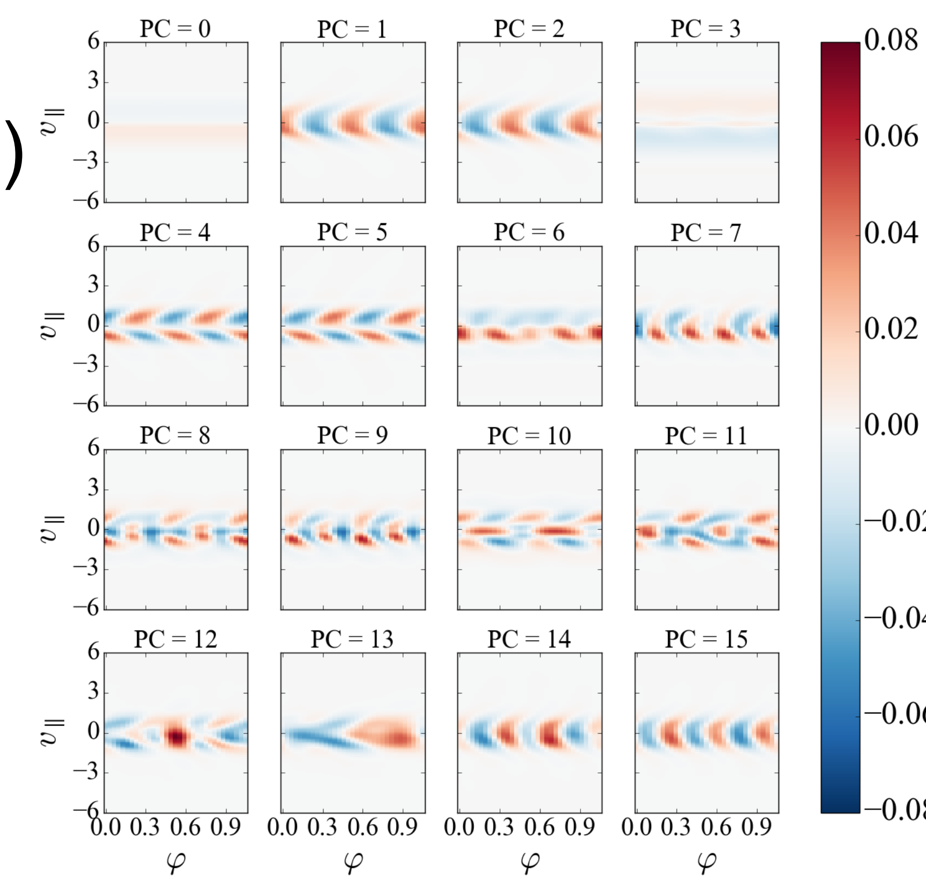
Coefficients



V^T

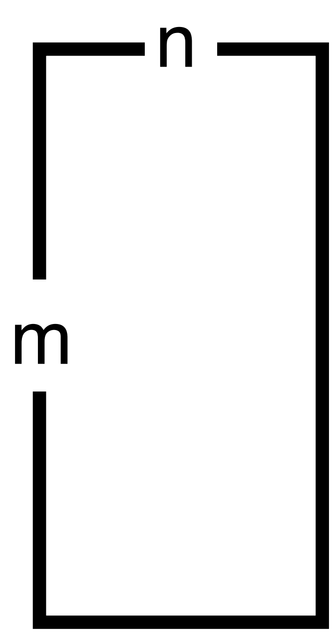
basis

(w = 0)



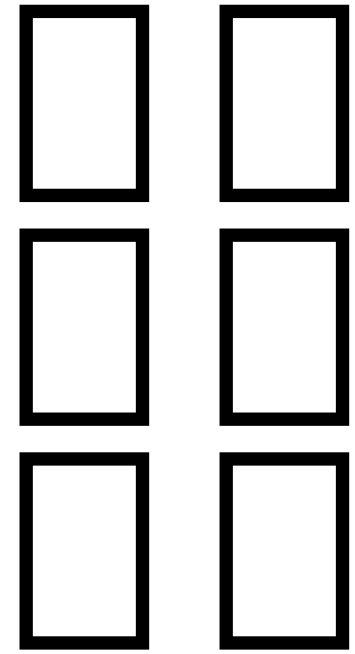
- Managing **out-of-memory data** (> 1TB) without MPI (Incremental PCA + randomized SVD)
- **Interpretable PCs:** 1/R dependency of B (PC 0), Ballooning modes (PC 1, 2)
- **Dask based** Incremental PCA merged to Dask-ml

Task level parallelization with Dask.distributed



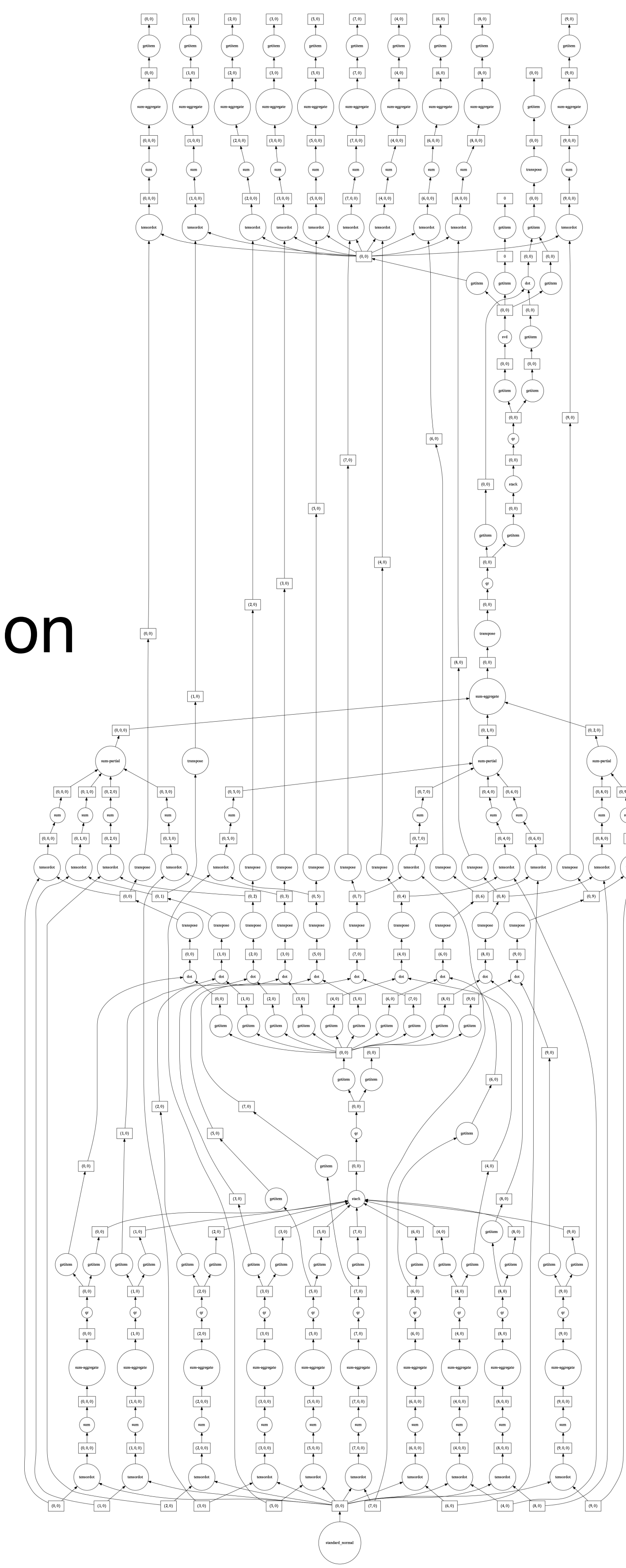
Large matrix
(out-of-memory)

DASK



Chunking into **on-memory** tasks
Process tasks in parallel based on dependency (by scheduler)

Task graph



```
X = da.random.random((10000, 100000), chunks=(1000, 100000)).persist()
cluster.scale(nb_workers)
client = Client(cluster)
start = time.time()
u, s, v = da.linalg.svd_compressed(X, k=4)
future = u.compute()
end = time.time()
```

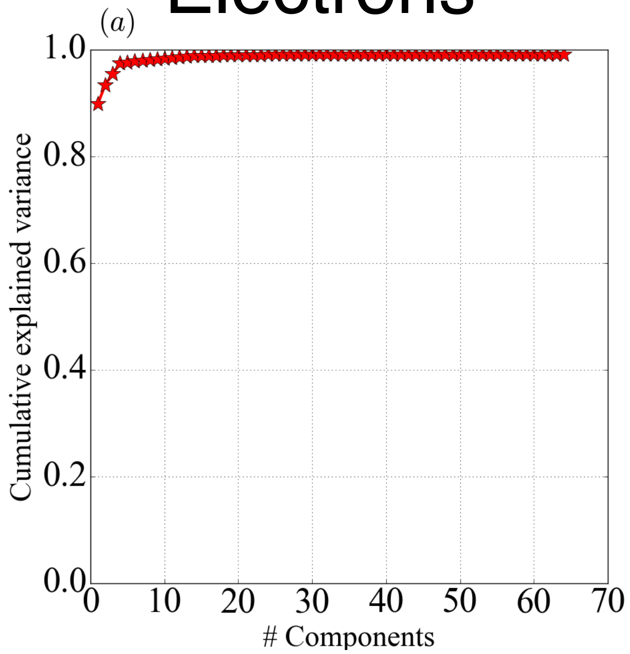
Chunking

```
It took 28.68612051010132  [s] with nb_workers 1, nb_cores 1
It took 20.172788381576538 [s] with nb_workers 2, nb_cores 1
It took 17.562381744384766 [s] with nb_workers 4, nb_cores 1
It took 12.512330770492554 [s] with nb_workers 8, nb_cores 1
```

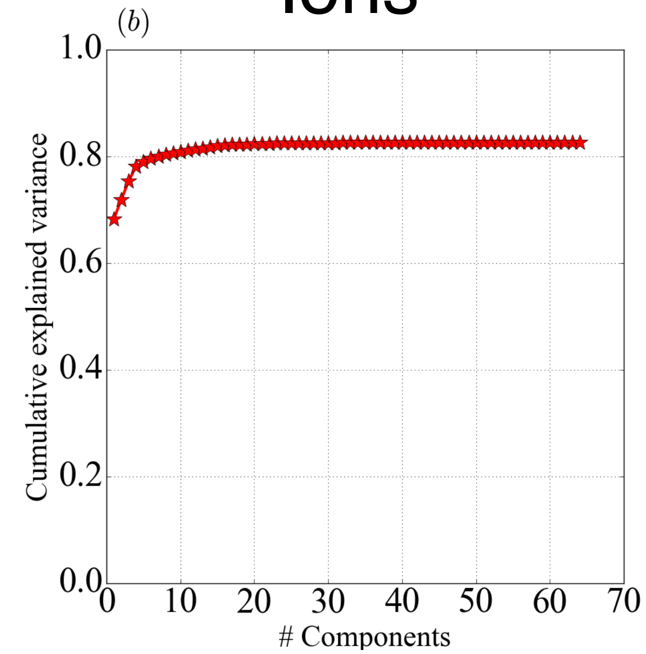
x 2.5 with 8 workers

Large scale PCA over 16 TB data

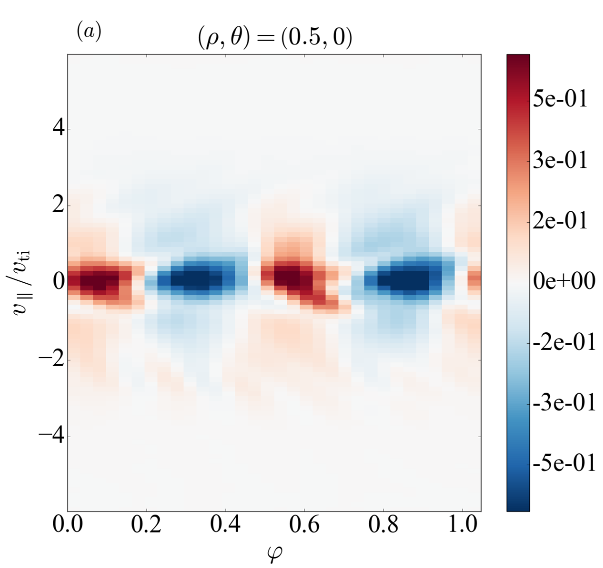
Electrons



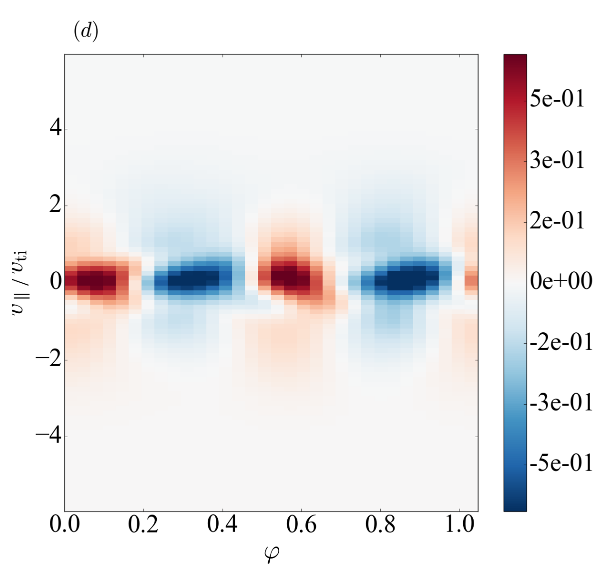
Ions



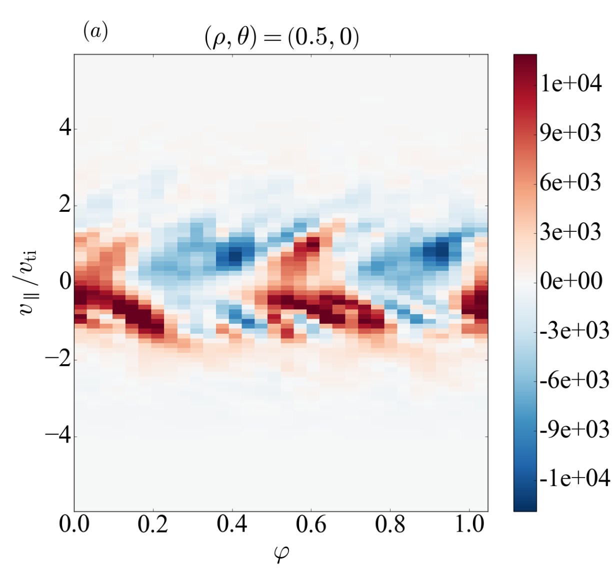
Reference



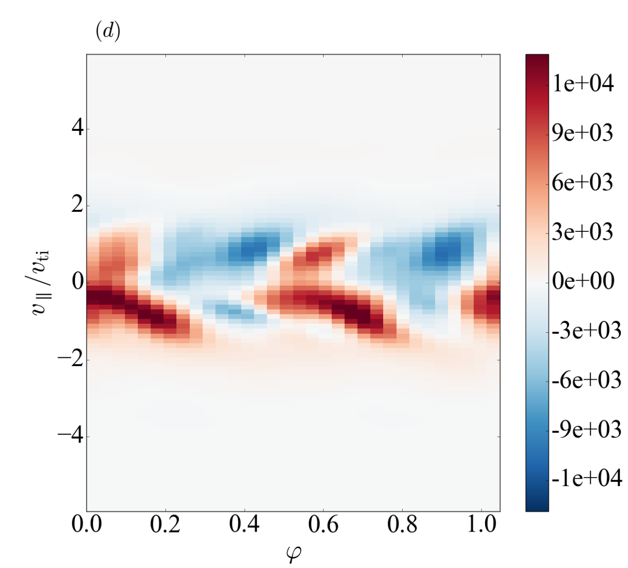
Reconstructed



Reference



Reconstructed



- Electron distribution function can be expressed with few components, while ion distribution function needs much more components
- **16 TB reduced into 7GB with 83 % of cumulative explained variance**

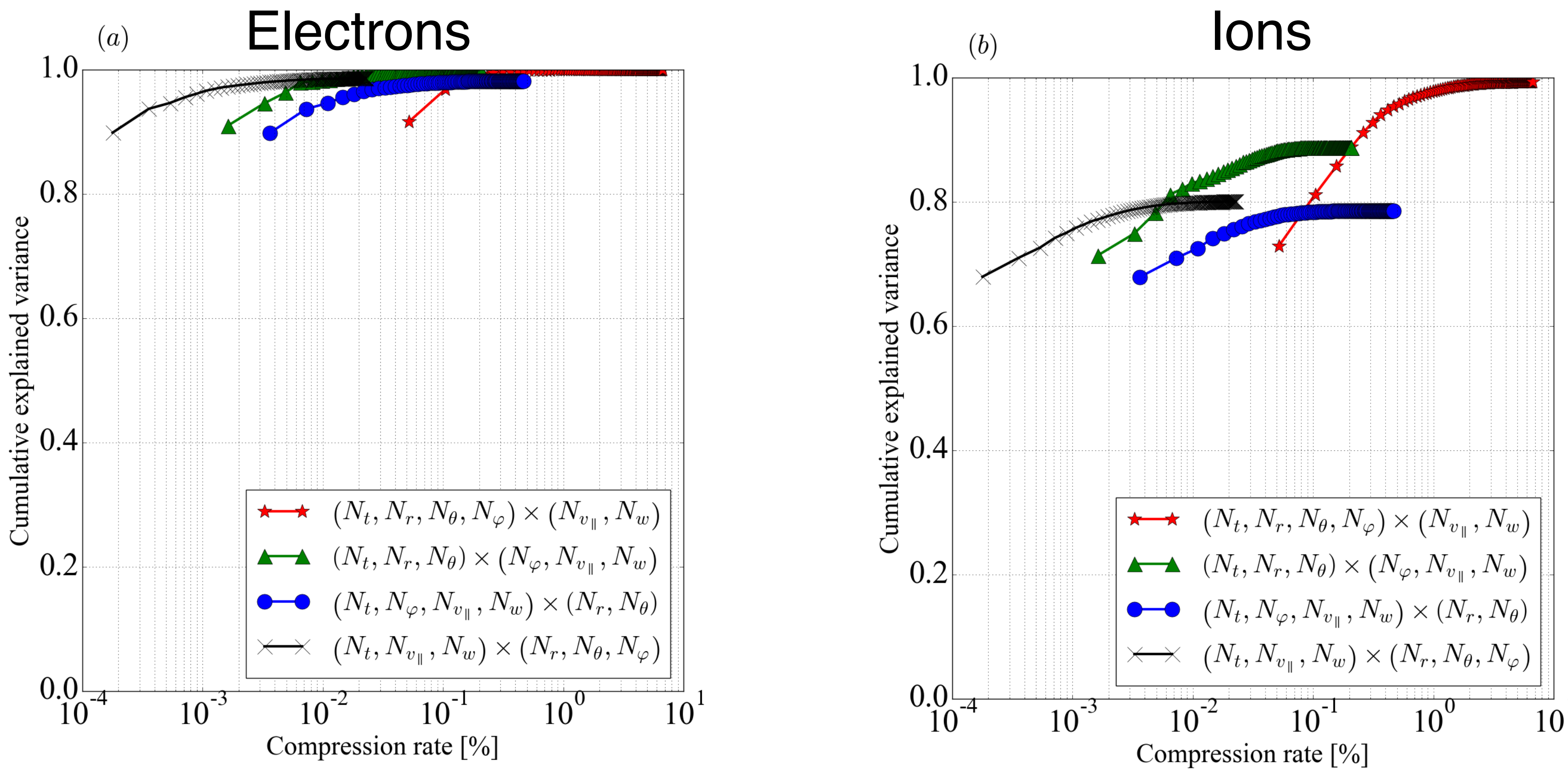
$$\underbrace{(\mathrm{time}, r, \theta)}_{\text{Samples}} \times \underbrace{(\varphi, v_\parallel, w)}_{\text{Features}}$$

Large scale PCA over 16 TB data

Previous version: random sampling from **a tiny part** of the entire data

Current version: **sampling from the entire data (turbulent part, 800 files)**

Higher: better representation



Smaller: stronger compression

Smaller: stronger compression

Electron distribution function can be expressed with few components, while ion distribution function needs much more components (**velocity space basis outperforms** [Hatch, JCP, 2012])

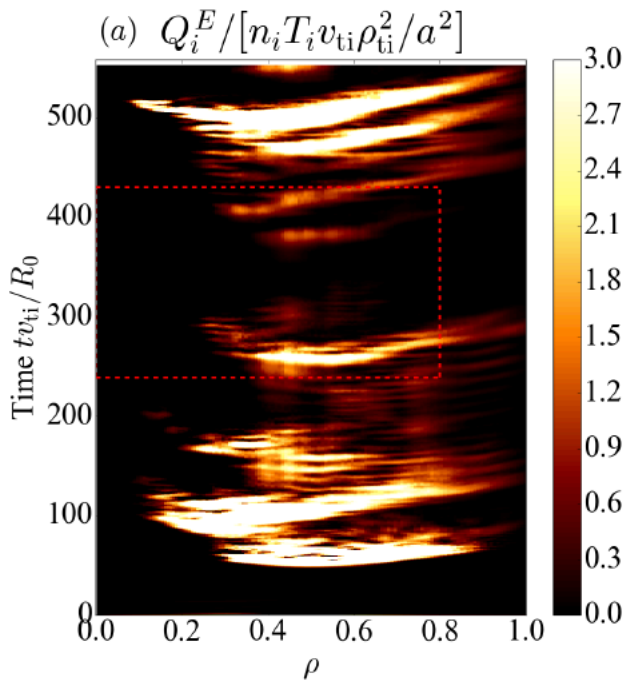
3 order of reduction in the data size

$$(\mathrm{time}, r, \theta) \times \left(\varphi, v_\parallel, w\right)$$

Samples Features

Energy flux recovered from reduced data

Reference



Energy flux by PCs

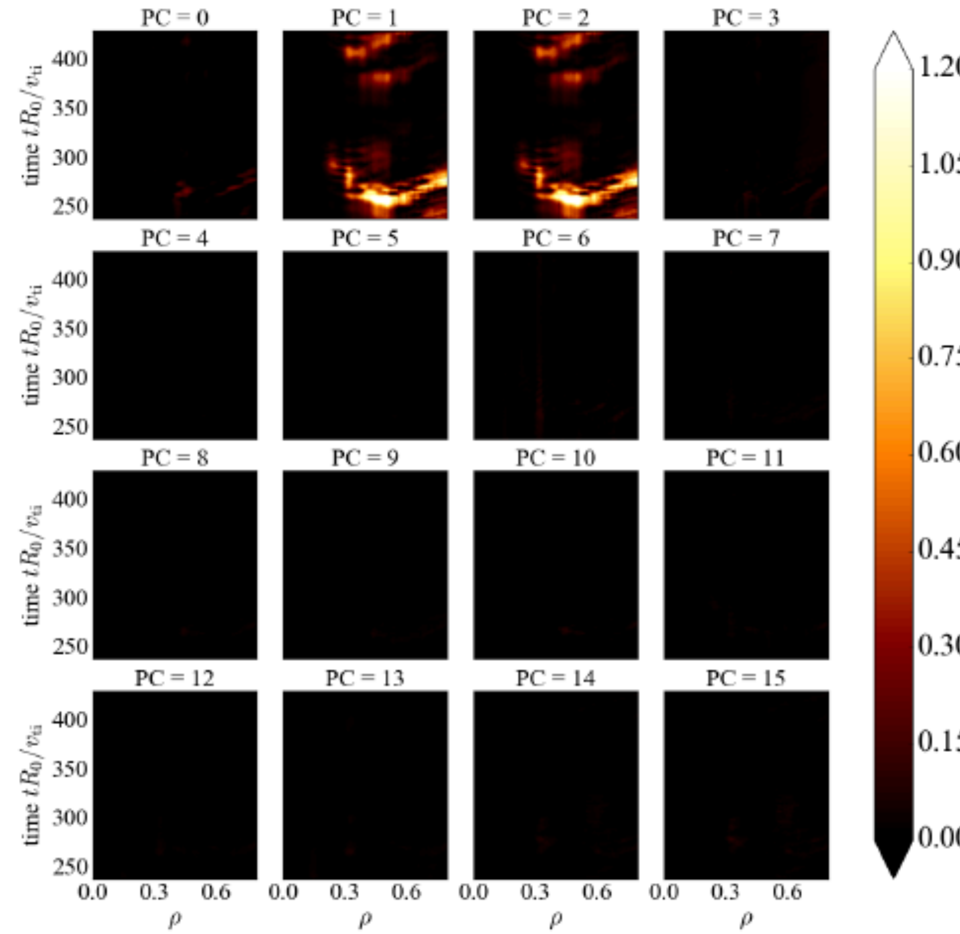


Figure 13: Spatio-temporal evolution of the reconstructed ion turbulent energy flux $\hat{Q}_i^E / [n_iT_iv_{\mathrm{ti}}\rho_{\mathrm{ti}}^2/a^2]$ driven by the first 16 principal components (PCs).

Approximated energy flux

$$
\begin{aligned}
\hat{Q}_i^E &= \int dv_\parallel d\mu 2\pi m_i^2 B_\parallel^* \left(\mathbf{v}_{E\times B}\cdot\nabla r\right)\left(\frac{m_i v_\parallel}{2}+\mu B\right)\hat{f} \\
&= \hat{Q}_{00} + \hat{Q}_{\mathrm{mean}} + \sum_j \hat{Q}_j,
\end{aligned}
$$

$$
\begin{aligned}
\hat{Q}_{00} &= \int dv_\parallel d\mu 2\pi m_i^2 B_\parallel^* \left(\mathbf{v}_{E\times B}\cdot\nabla r\right)\left(\frac{m_i v_\parallel}{2}+\mu B\right) f_{00} \\
\hat{Q}_{\mathrm{mean}} &= \int dv_\parallel d\mu 2\pi m_i^2 B_\parallel^* \left(\mathbf{v}_{E\times B}\cdot\nabla r\right)\left(\frac{m_i v_\parallel}{2}+\mu B\right) \overline{f} \\
\hat{Q}_j &= \int dv_\parallel d\mu 2\pi m_i^2 B_\parallel^* \left(\mathbf{v}_{E\times B}\cdot\nabla r\right)\left(\frac{m_i v_\parallel}{2}+\mu B\right) \mathbf{p}_j x_j
\end{aligned}
$$

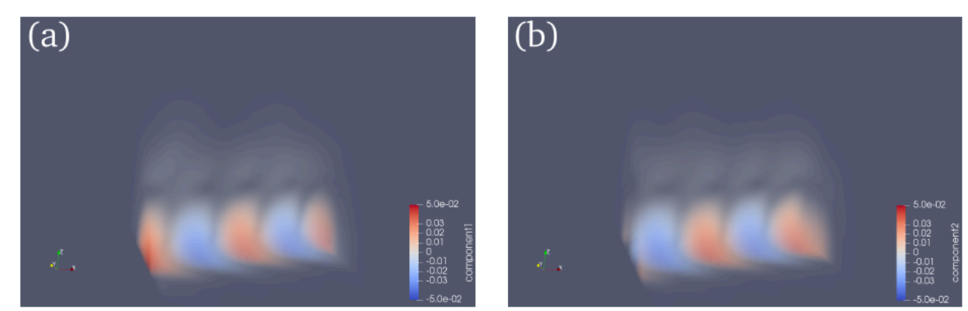


Figure 14: The principal components 1 (a) and 2 (b) of phase space bases $\mathbf{p}_j$. The directions x, y, and z correspond to the directions φ, $v_\parallel$, and w, respectively.

- 3 order reduction (DOF: 10^12 to 10^9) of the data size, still keeping the important properties like avalanche like transport

3 levels of Postscripting

Postprocesing the completed simulation data on the disk [DONE]

```
for it in range(nb_iter):
    phi = xr.open_dataset(all_existing_files[it])
    phi = preprocess(phi)
    ipca.partial_fit(data=phi, it=it)
    it_start += 1
```

Loading the data from a **file already existing**

> 10 TB storage

Postprocesing the on-going simulation data on the disk [DONE]

```
for it range(nb_iter):
    ready = result_monitor.wait(it_start=it_start)
    if ready:
        phi = result_monitor.get(it=it)
        phi = preprocess(phi)
        ipca.partial_fit(data=phi, it=it)
        it_start += 1
```

Loading the data from a **file as soon as generated**

< 100 GB storage

Postprocesing the on-going simulation data on the **memory**

```
for it range(nb_iter):
    ready = result_monitor.wait(it_start=it_start)
    if ready:
        phi = result_monitor.get(it=it)
        phi = preprocess(phi)
        ipca.partial_fit(data=phi, it=it)
        it_start += 1
```

Loading the data on **memory through PDI + Dask**

< No extra storage

With the help of Dr. J. Bigot

Objectives

Scalable Data analysis

- Large scale data analysis based on Dask
- Analyzing the time series of 5D distribution function [1] ★
- In-situ machine learning to avoid saving the huge data

AI 4 CFD

- Convolutional neural network to predict the multi-resolution steady flow data [2]
- Data-driven modeling of Subgrid scale dynamics

Data driven analyses for Exascale simulations

[1] Y. Asahi et al., Phys. Plasmas, 2021
[2] Y. Asahi et al., to be submitted

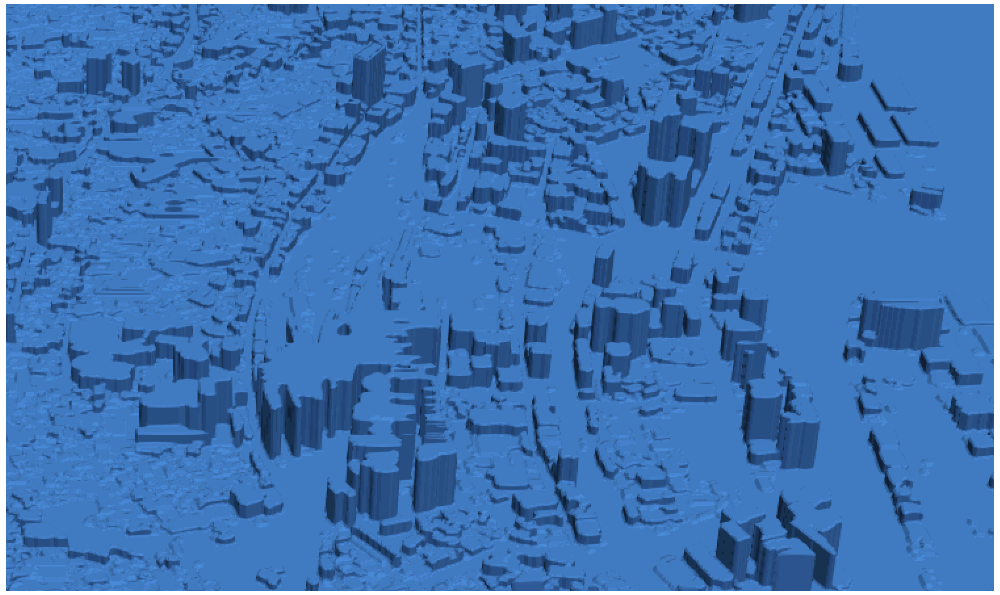
★ Completed in JH200065 "Modernizing and accelerating fusion plasma turbulence codes targeting exa-scale systems"

Motivation and objectives

Motivation

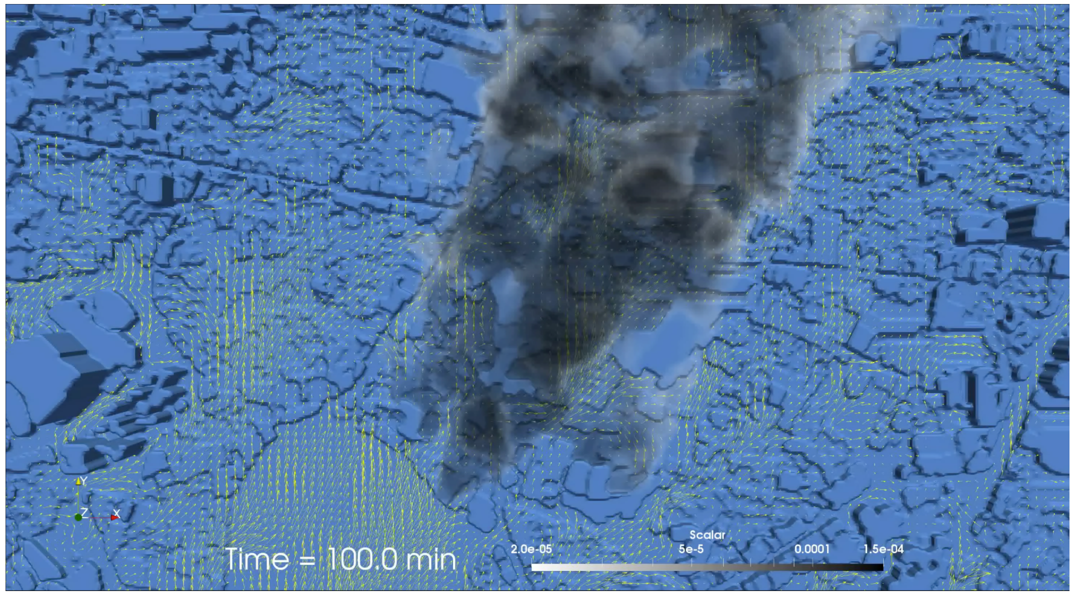
- High resolution fluid simulations are getting more and more costly
- Surrogate models based on Deep learning methods can be used to predict steady flows from signed distance functions
- It is difficult to apply these DL models to high or multi-resolution data, particularly when the data are given in a distributed manner

Input: Signed distance function (SDF)

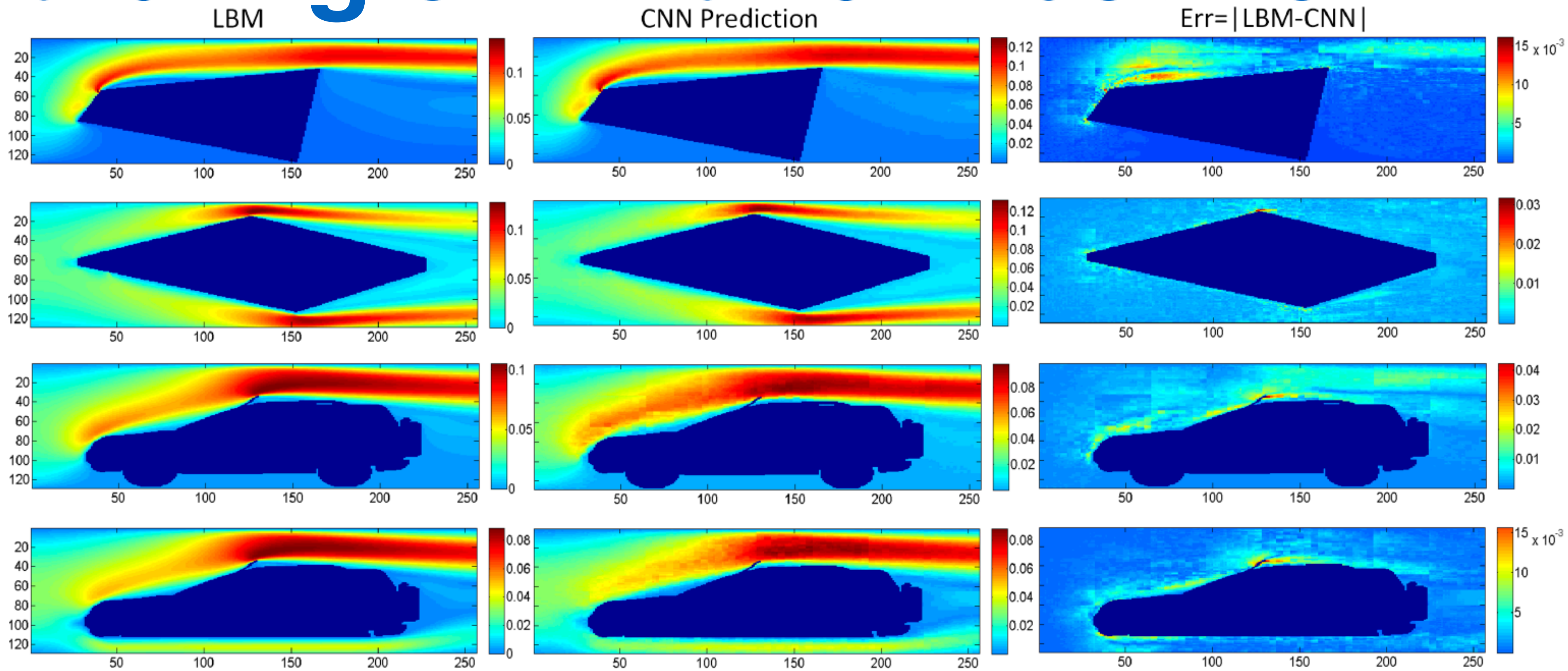


CNN

Output: flow fields



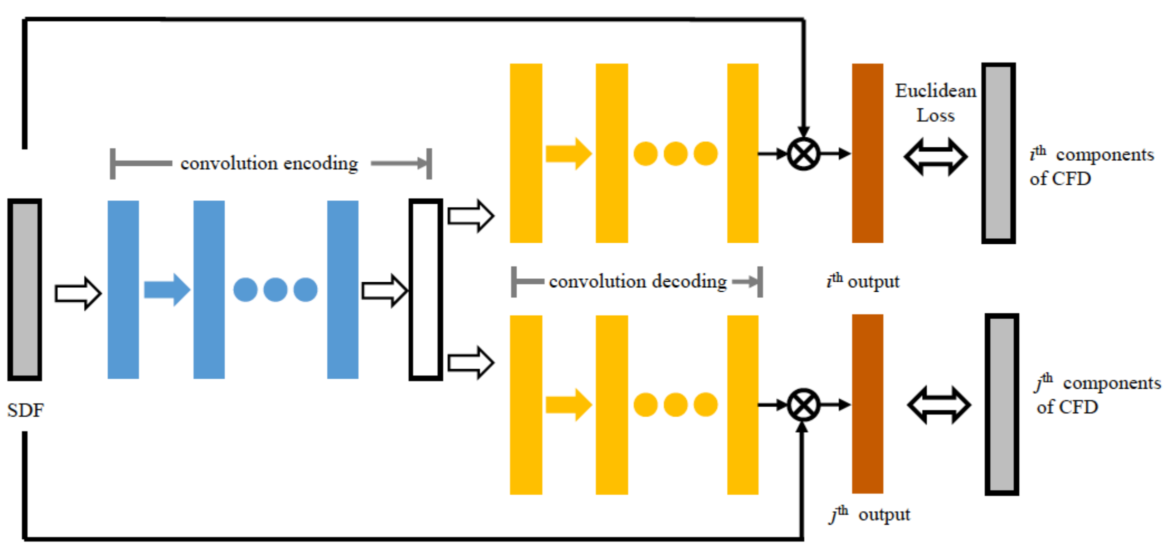
Predicting simulation results with DL



CNN model (Guo et al, 2016)

Input (signed distance function (SDF)): H x W

Output (flow fields): H x W x D



Boundary: SDF == 0

Inside an object: SDF < 0

Outside an object: SDF >0

Guo et al, 2016

Predicting steady flows in uniform unpatched grid from SDF

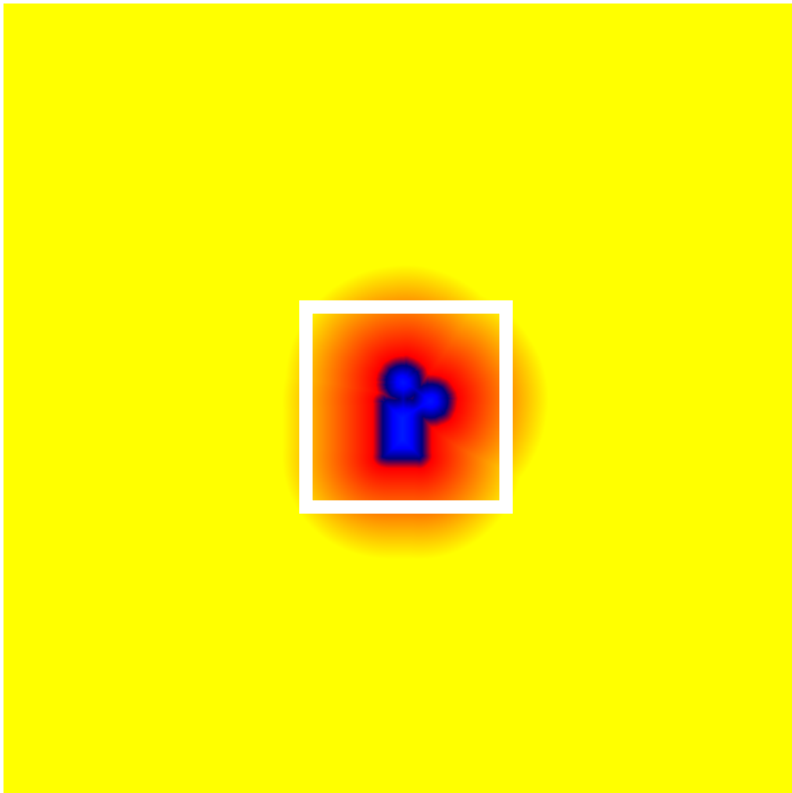
This work

Predicting steady flows in Adaptive Mesh Refinement (AMR) grid from SDF

2D AMR dataset with LBM calculations

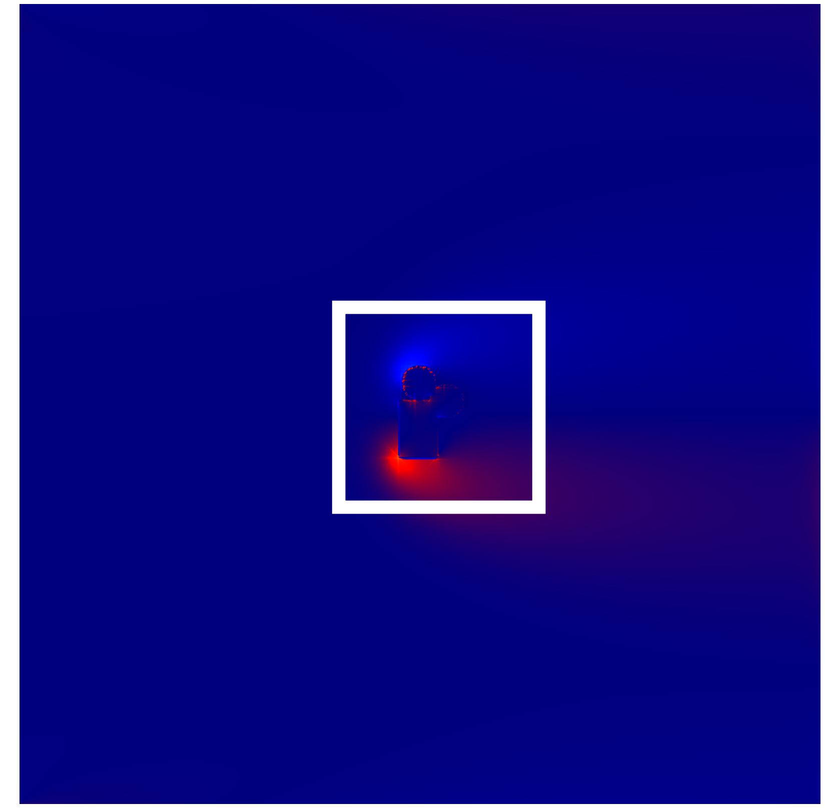
SDF (1024x1024)

Inflow →



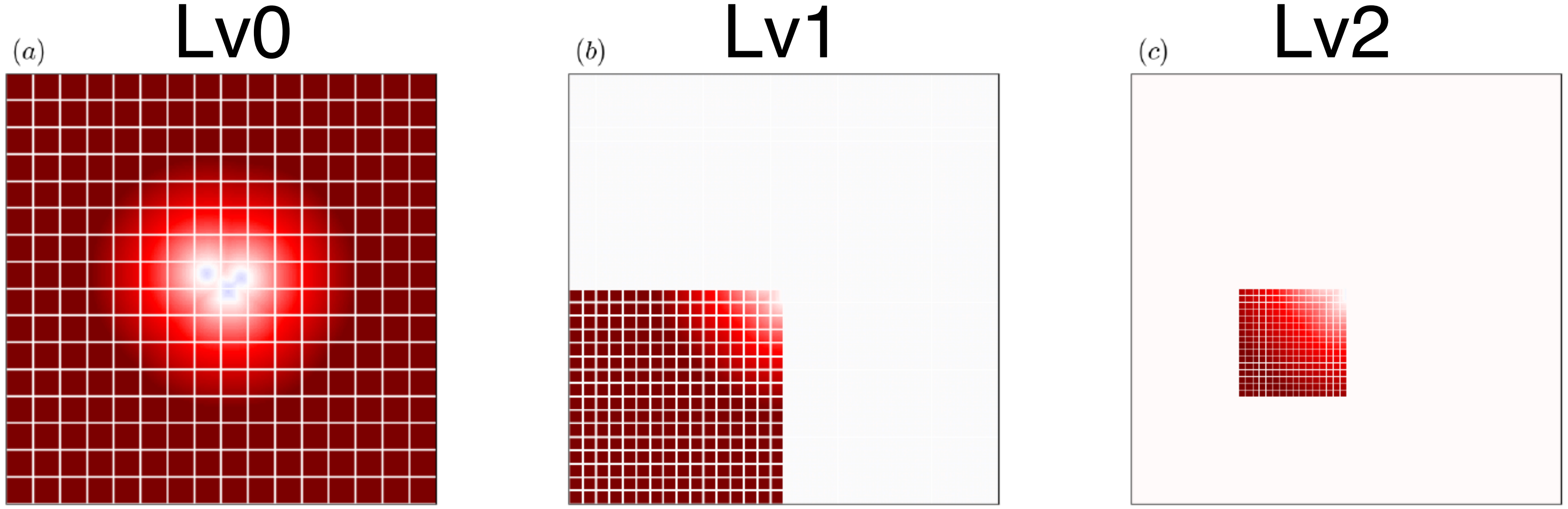
Outflow →

Vorticity (1024x1024)



2D simulation: putting 1-5 objects located around the center region (256x256)

SDF



Data

train:	9500 shots
validation:	250 shots
test:	250 shots

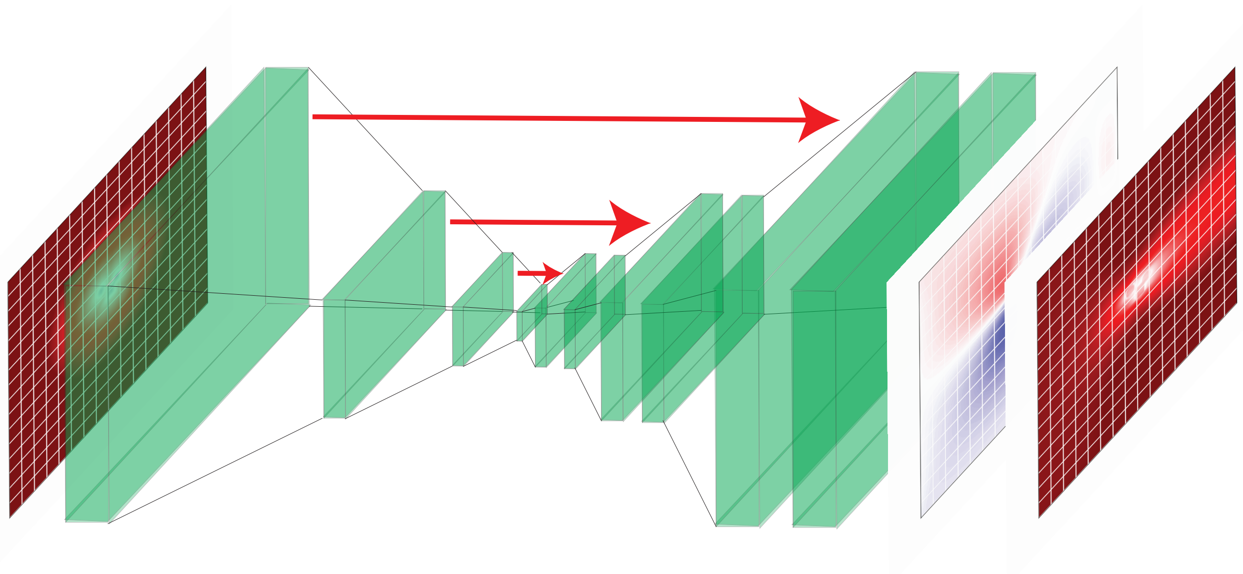
Lv0: (1 x 1 x 256 x 256), (256 x 256 low resolution、un-patched)

Lv1: (2 x 2 x 256 x 256), (512 x 512 middle resolution、**patched**)

Lv2: (4 x 4 x 256 x 256), (1024 x 1024 high resolution、**patched**)

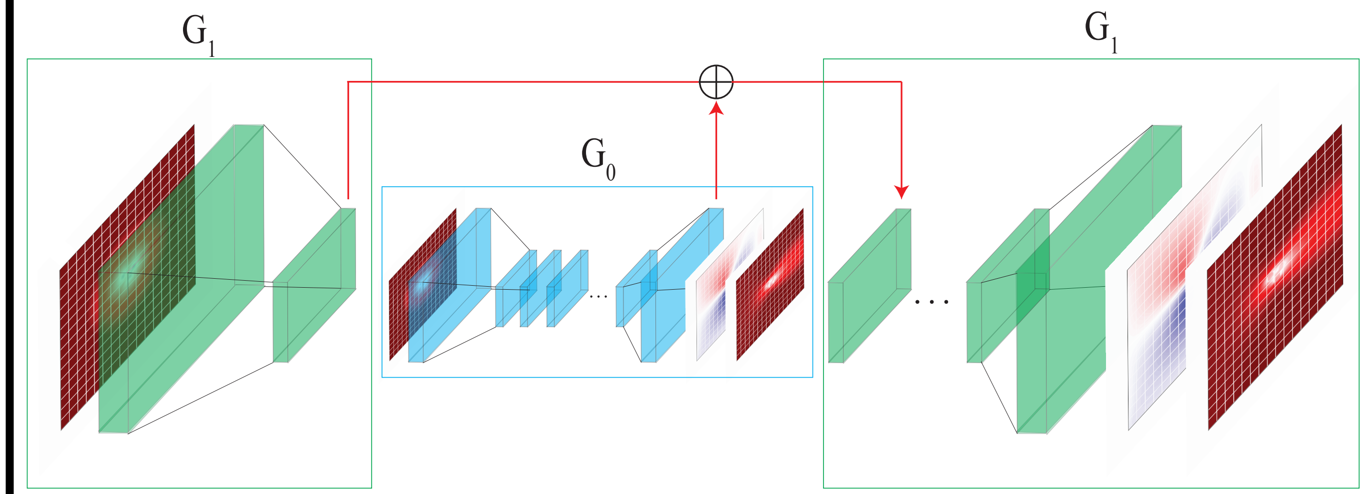
CNN architectures (SDF→flows)

UNet [1]



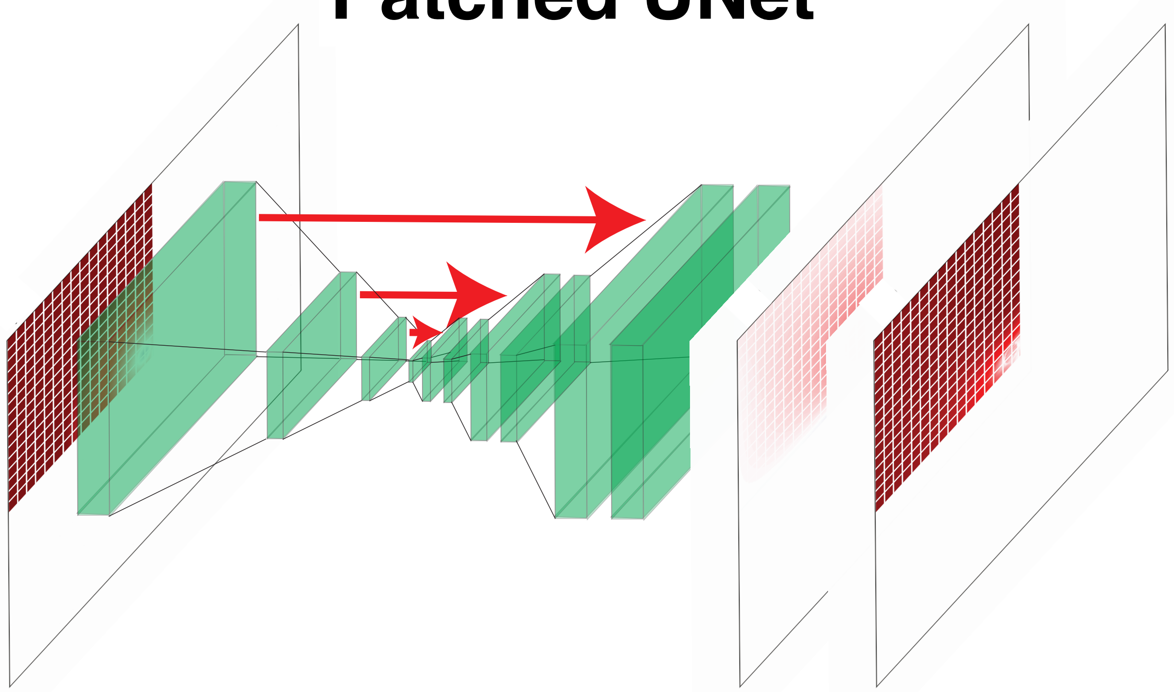
Basic Unet

Pix2PixHD [2]



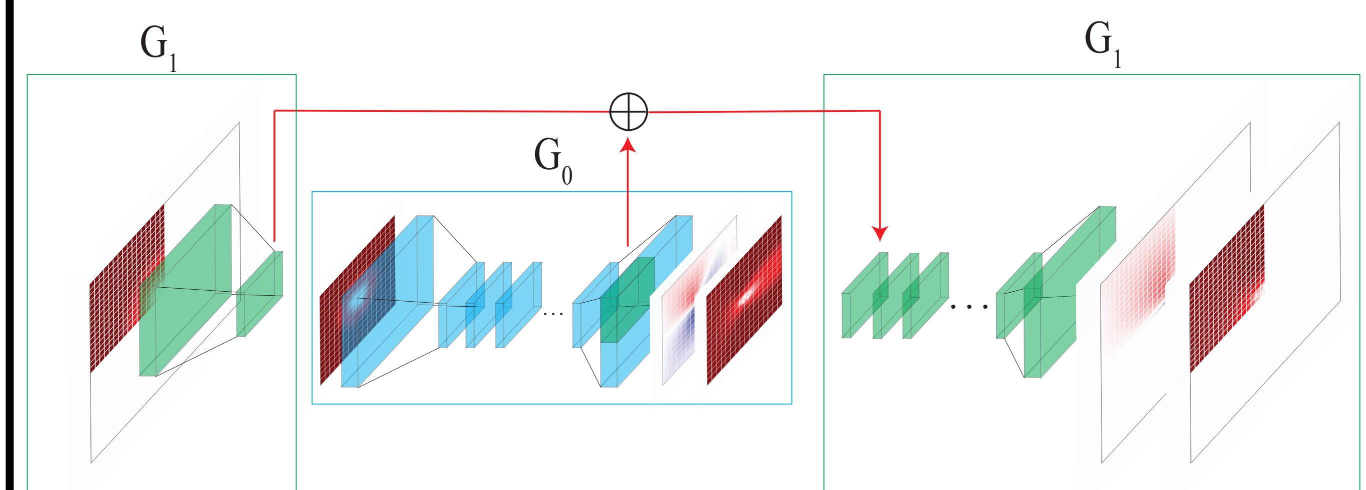
ResNet architecture
Global generator + local enhancer

Patched UNet



Unet applied to patched data

Patched Pix2PixHD

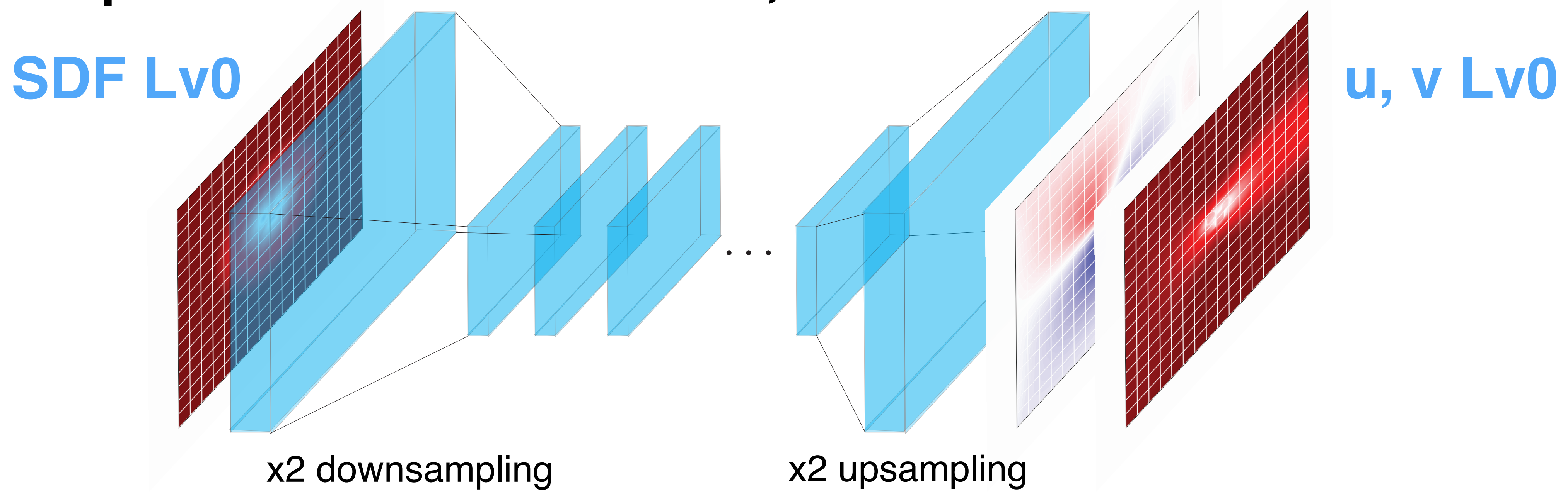


Pix2PixHD applied to patched data

[1] Olaf Ronneberger, 2015 [2] Ting-Chun Wang, 2018

Pix2PixHD [1] architecture

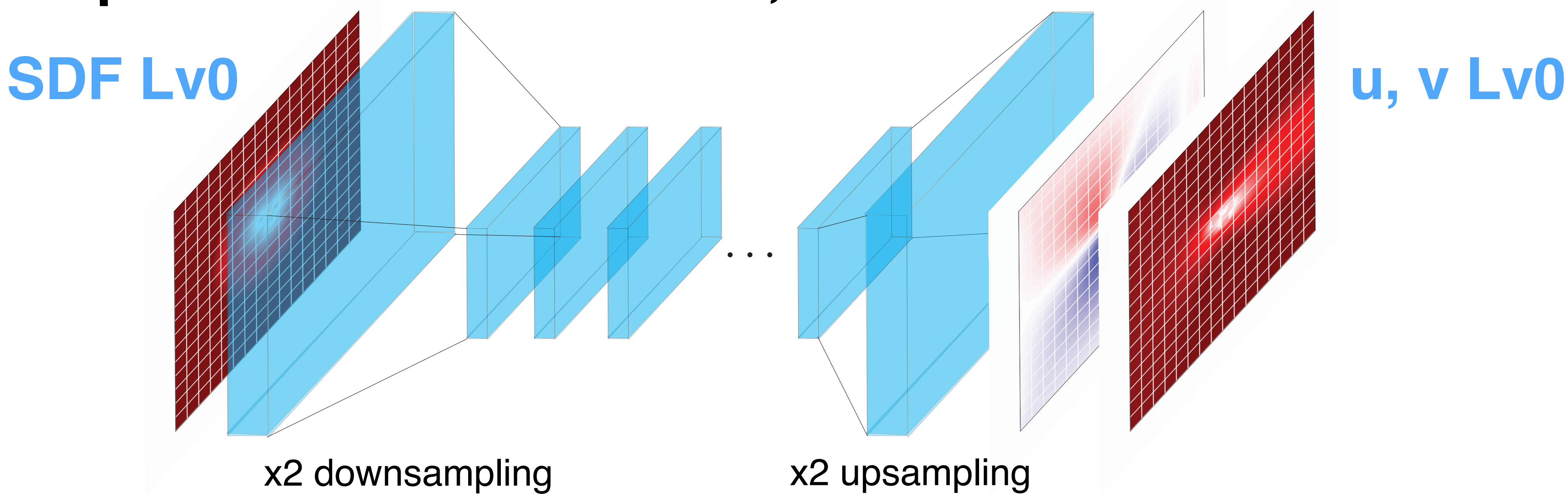
Lv0 prediction: SDF Lv0 -> u, v Lv0



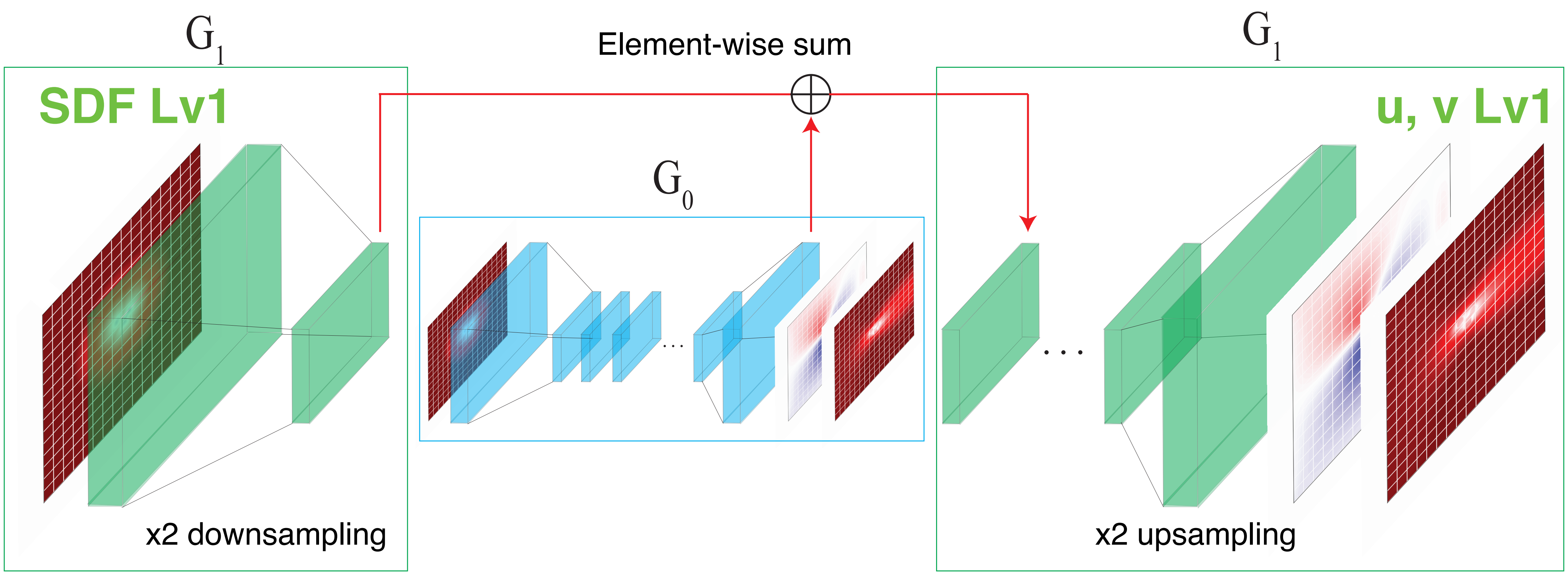
[1] Ting-Chun Wang, 2018

Pix2PixHD [1] architecture

Lv0 prediction: SDF Lv0 -> u, v Lv0



Lv1 pred: SDF Lv1 （high res.） + SDF Lv0 -> u, v Lv1 （high res.）

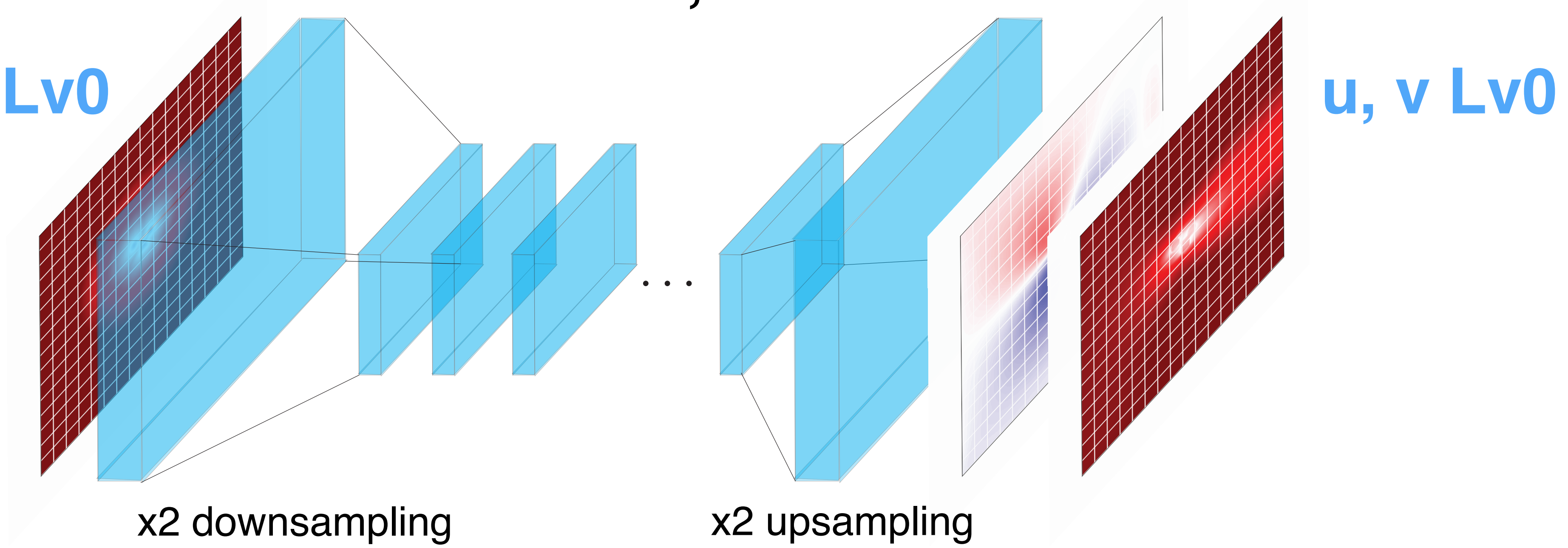


G_0: Global generator
G_1: Local enhancer

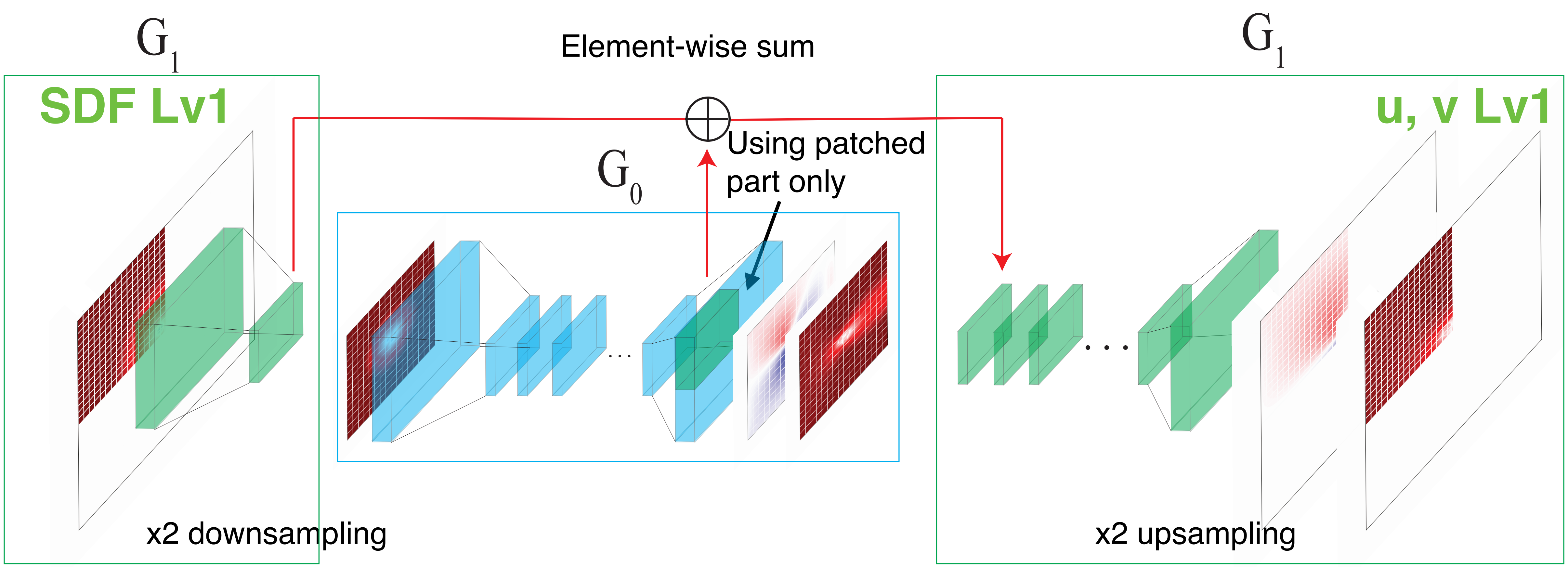
[1] Ting-Chun Wang, 2018

AMR Net (patched Pix2PixHD) architecture

Lv0 prediction: SDF Lv0 -> u, v Lv0



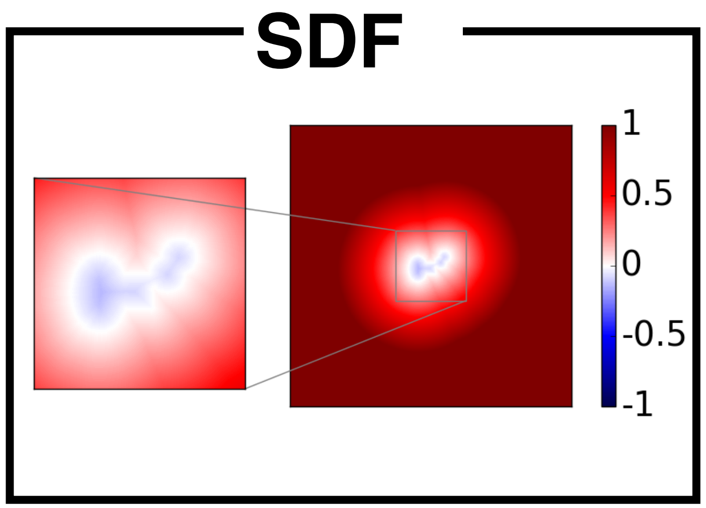
Lv1 prediction: SDF Lv1 (patched) + SDF Lv0 -> u, v Lv1 (patched)



patched data only for high resolution→memory efficient

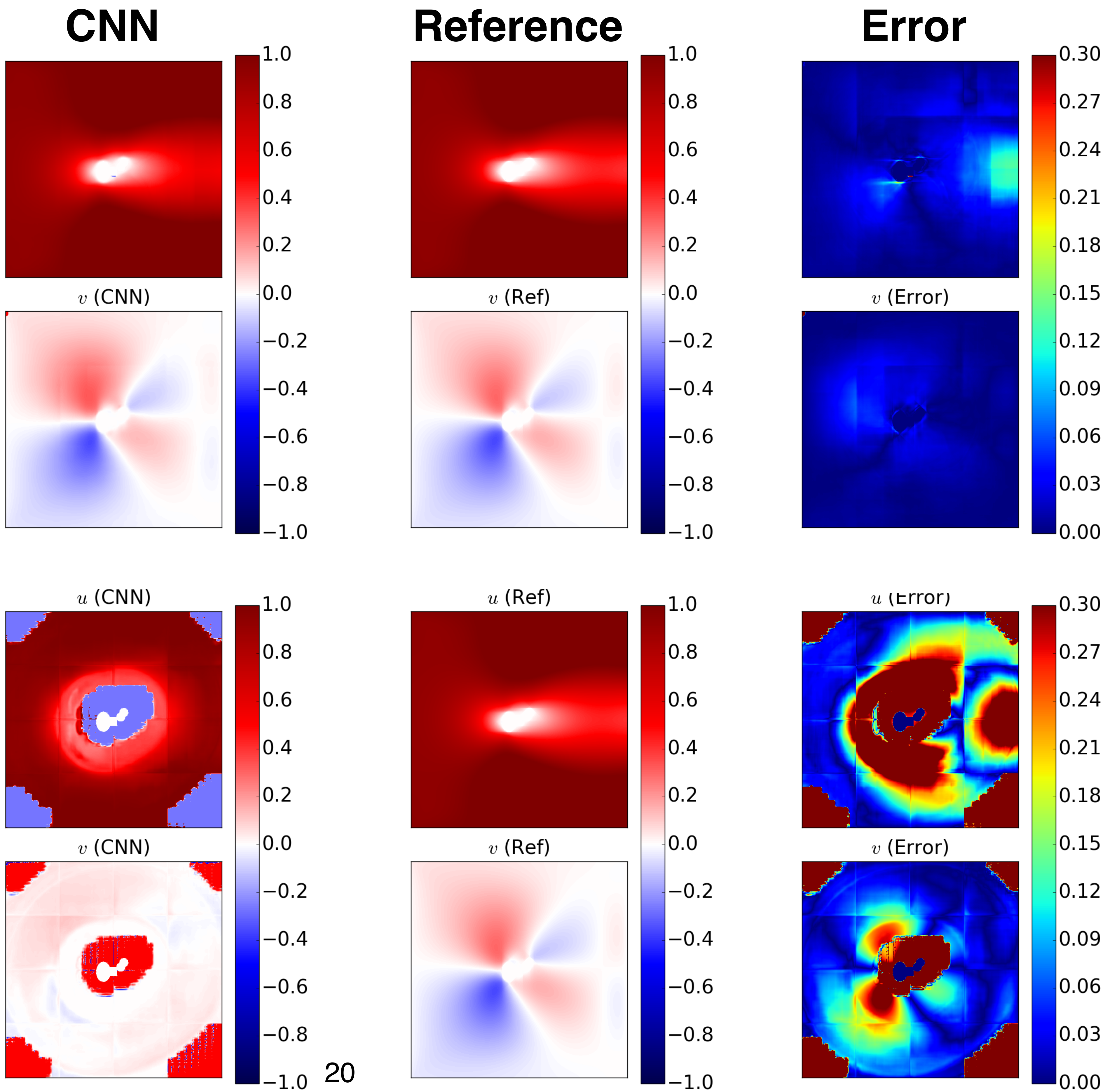
U-Net cannot predict flows from patched data

UNet



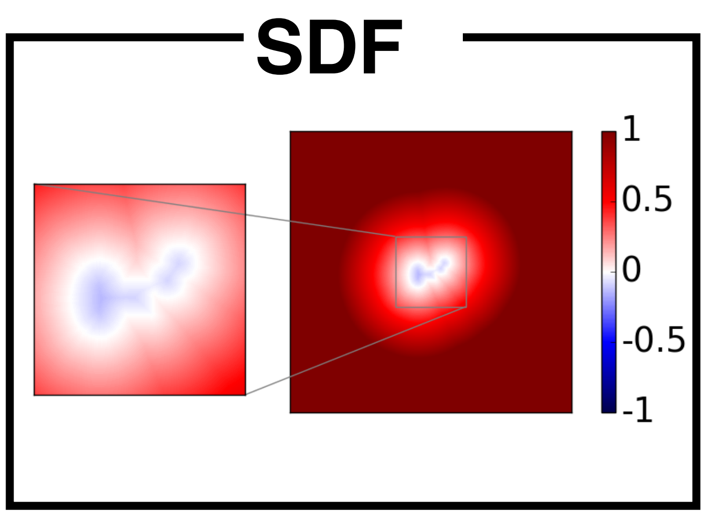
UNet (patched)

Lv2 (1024x1024) prediction (test data)



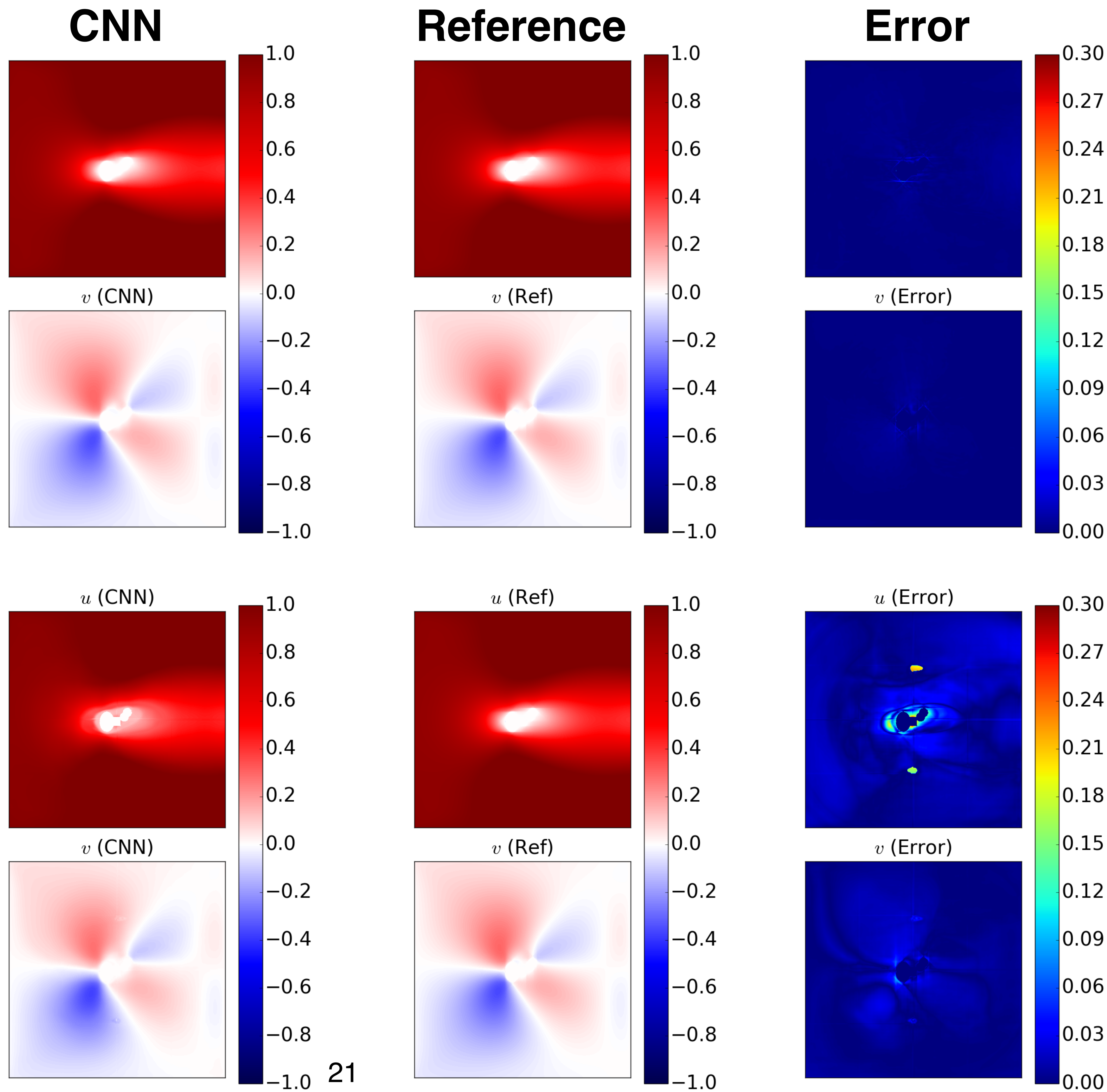
AMRNet can predict global flows from patched data

Pix2PixHD



Pix2PixHD (patched)

Lv2 (1024x1024) prediction (test data)



Summary

Scalable data analysis based on Dask

- Extereme scale PCA on the time series of 5D data
- Integrate Dask into GYSELA diags through PDI (developed by J. Bigot)
- In-situ PCA on GYSELA data without saving data on a disk

AMR-Net: CNN model for Multi-resolution steady flow prediction

- Pix2PixHD based model to predict global flows from patched data
- Suppress memory usage and applicable to a distributed dataset

Target Conferences/Workshop

- HP-C/DA in ISC
- AI4S in IEEE Cluster