StyleGAN as an AI Deconvolution Operator for Large Eddy Simulations of Turbulent Plasma Equations in BOUT++

Project: FARSCAPE II

Jony Castagna – UKRI-STFC Hartree Centre

Francesca Schiavello – UKRI-STFC Hartree Centre

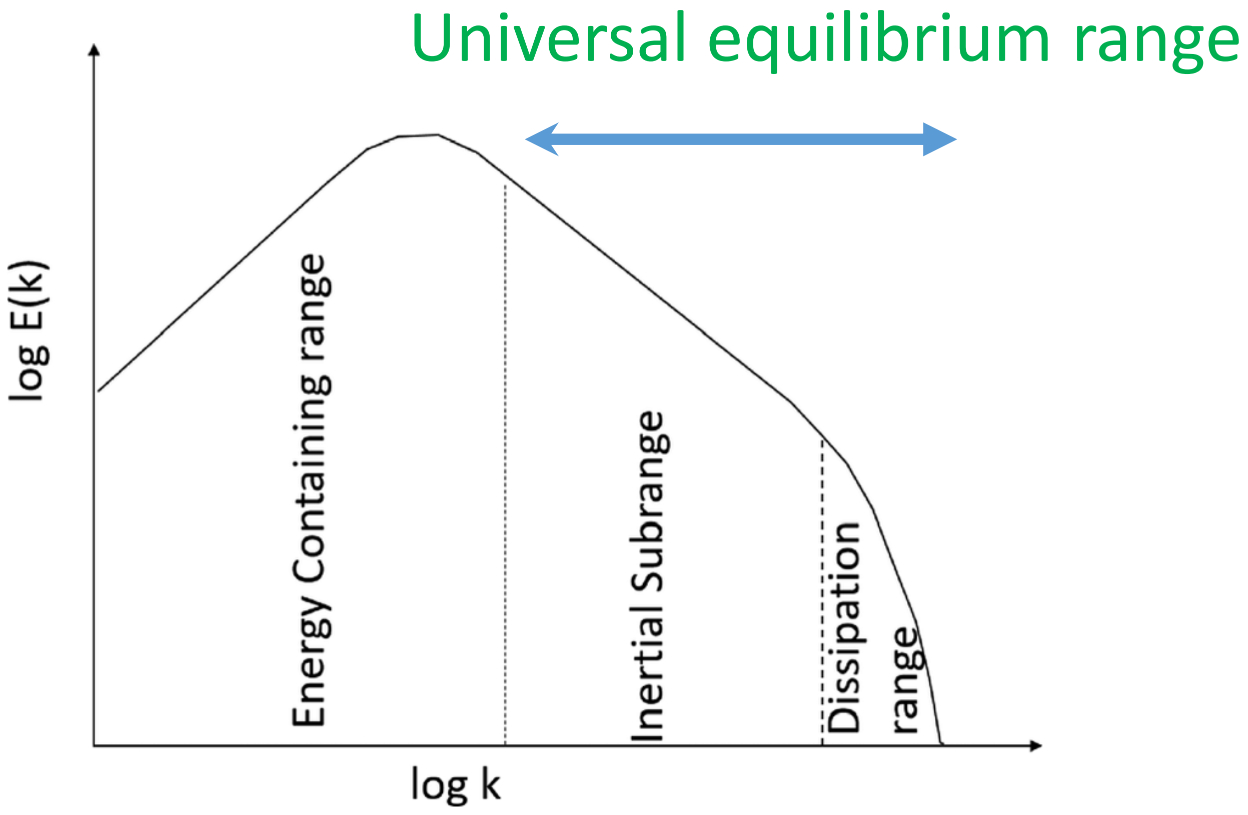
BOUT++ meeting 2023: 09-13/01/2023

UKRI Science and Technology Facilities Council
Hartree Centre

Outlines

- Large Eddy Simulation (LES)
- GAN and StyleGAN
- The StylES procedure
- Results on the Hasegawa Wakatani (HW) test case
- Integration with BOUT++
- Resume

Large Eddy Simulation (I)



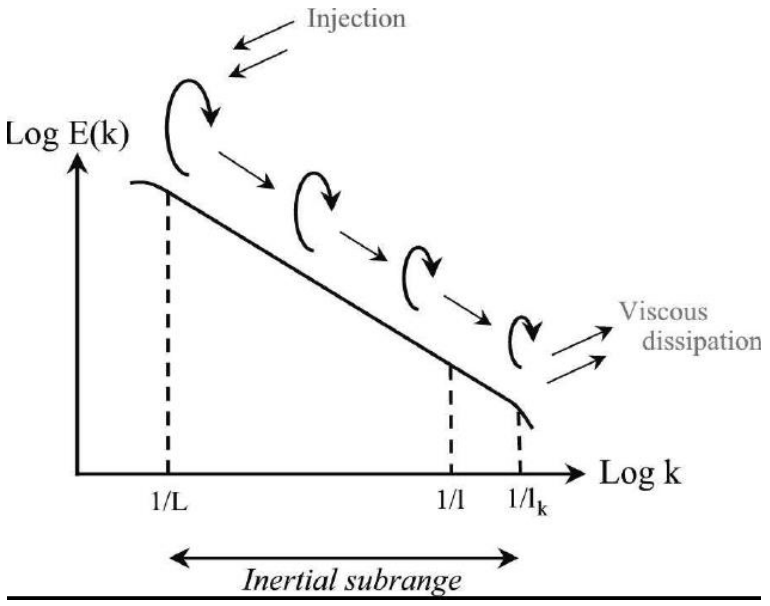
Kolmogorov theory:

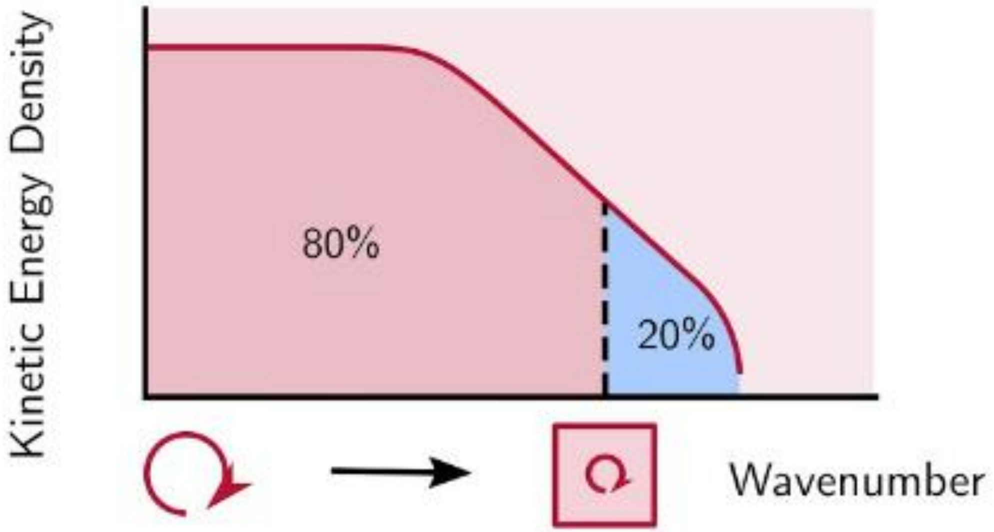
1) At large Re number the small scales of turbulence are isotropic and have a **universal structure** (i.e. independent of the flow)

2) At large Re the local average properties of the small- scale components of any turbulent motion are determined entirely by kinematic viscosity and average rate of dissipation per unit mass.

3) There is an upper subrange (the inertial subrange) in which the local average properties are determined only by the rate of dissipation per unit mass.

Large Eddy Simulation (II)





In DNS we want to solve the full range of scales

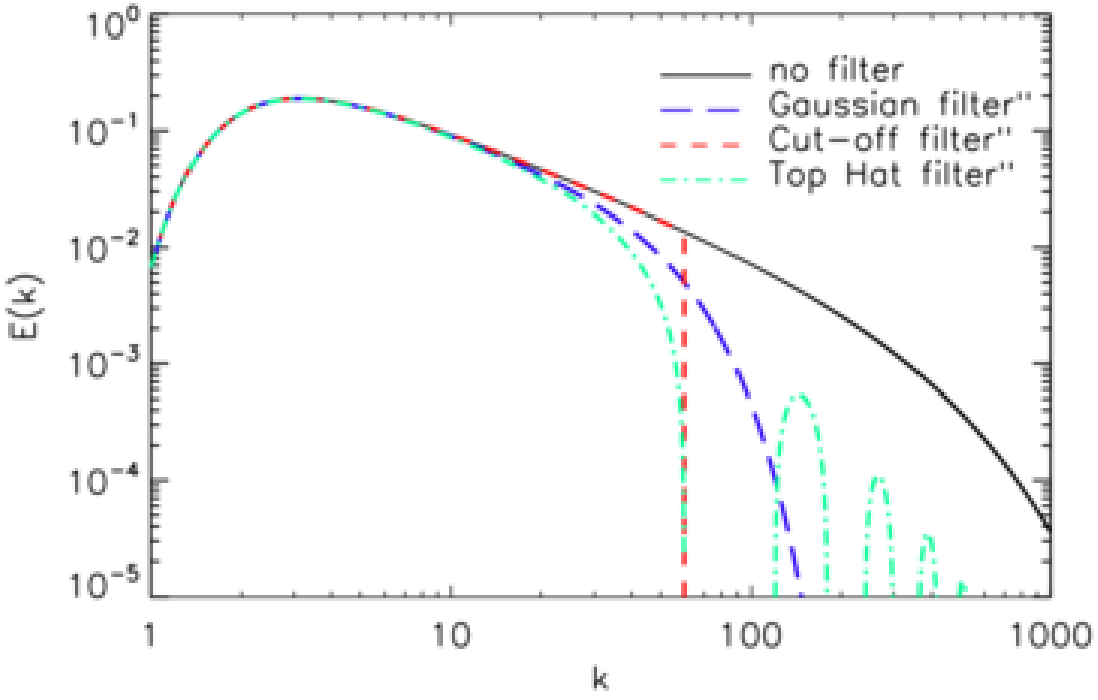
In LES we want to solve only up to a certain range

LES is much faster than DNS but needs modelling the sub-grid scale tensor!

Science and Technology Facilities Council

Hartree Centre

Large Eddy Simulation (III)



$$\overline{\phi(\boldsymbol{x},t)} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi(\boldsymbol{r},t')G(\boldsymbol{x}-\boldsymbol{r},t-t')dt'd\boldsymbol{r},$$

$$\overline{\phi} = G \star \phi.$$ Filter operator

N-S equation

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\nu \frac{\partial u_i}{\partial x_j}\right)$$

Filter

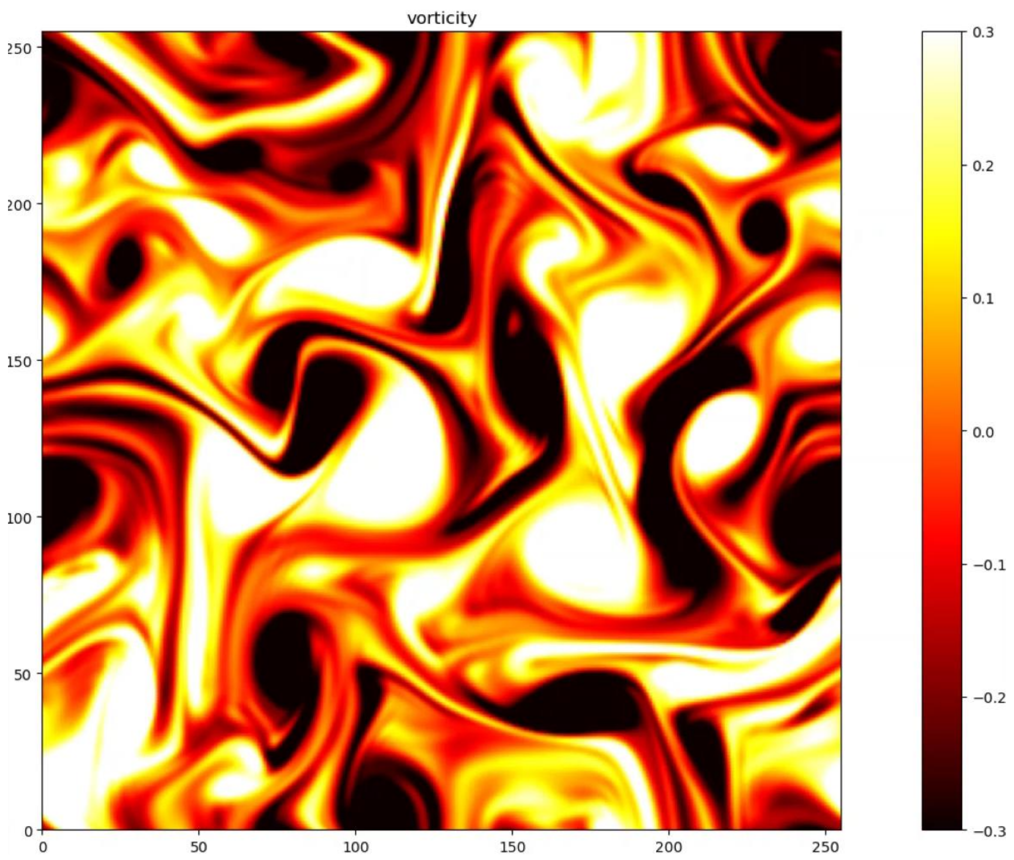
Filtered N-S equation

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial \overline{u}_i \overline{u}_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\nu \frac{\partial \overline{u}_i}{\partial x_j}\right) - \frac{\partial \tau_{ij}}{\partial x_j}$$

$$\boxed{\tau_{ij} \equiv \overline{u_i u_j} - \overline{u}_i \overline{u}_j} \longleftarrow \text{Needs modeling}$$

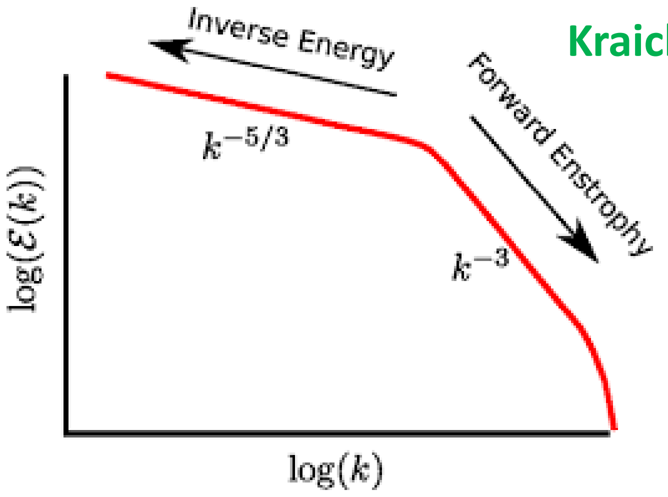
Sub-grid scale (SGS) stress

2D Homogeneous Isotropic Turbulent (2D-HIT)

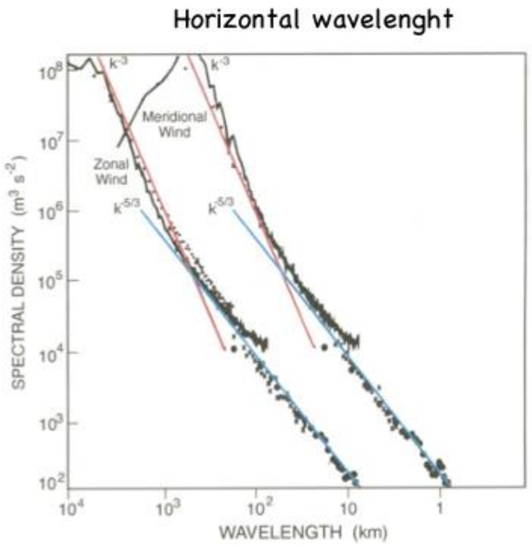


Vorticity field

Kraichnan 1967 theory



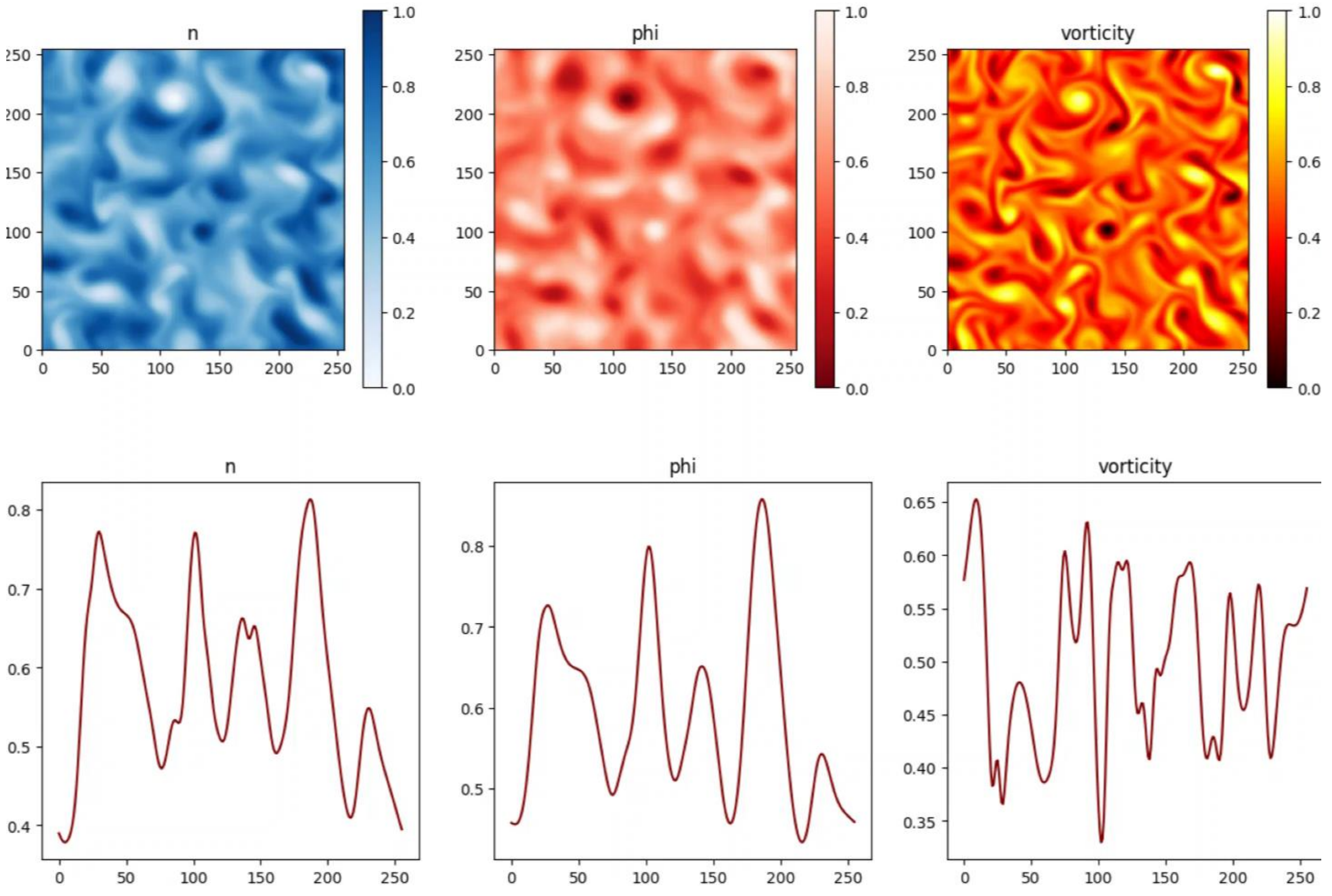
Inverse cascade of energy



Nastrom , Gage, Jasperson , Nature 310 (1984)

Experimental evidences from atmospheric turbulence measurements!

Hasagawa Wakatani (HW)



Results obtained with BOUT++

2D Navier-Stokes equations

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)$$

Similar fluid dynamic behaviours!

$$\frac{\partial \zeta}{\partial t} + \{\phi, \zeta\} = \alpha(\phi - n) - \mu\nabla^4\zeta$$

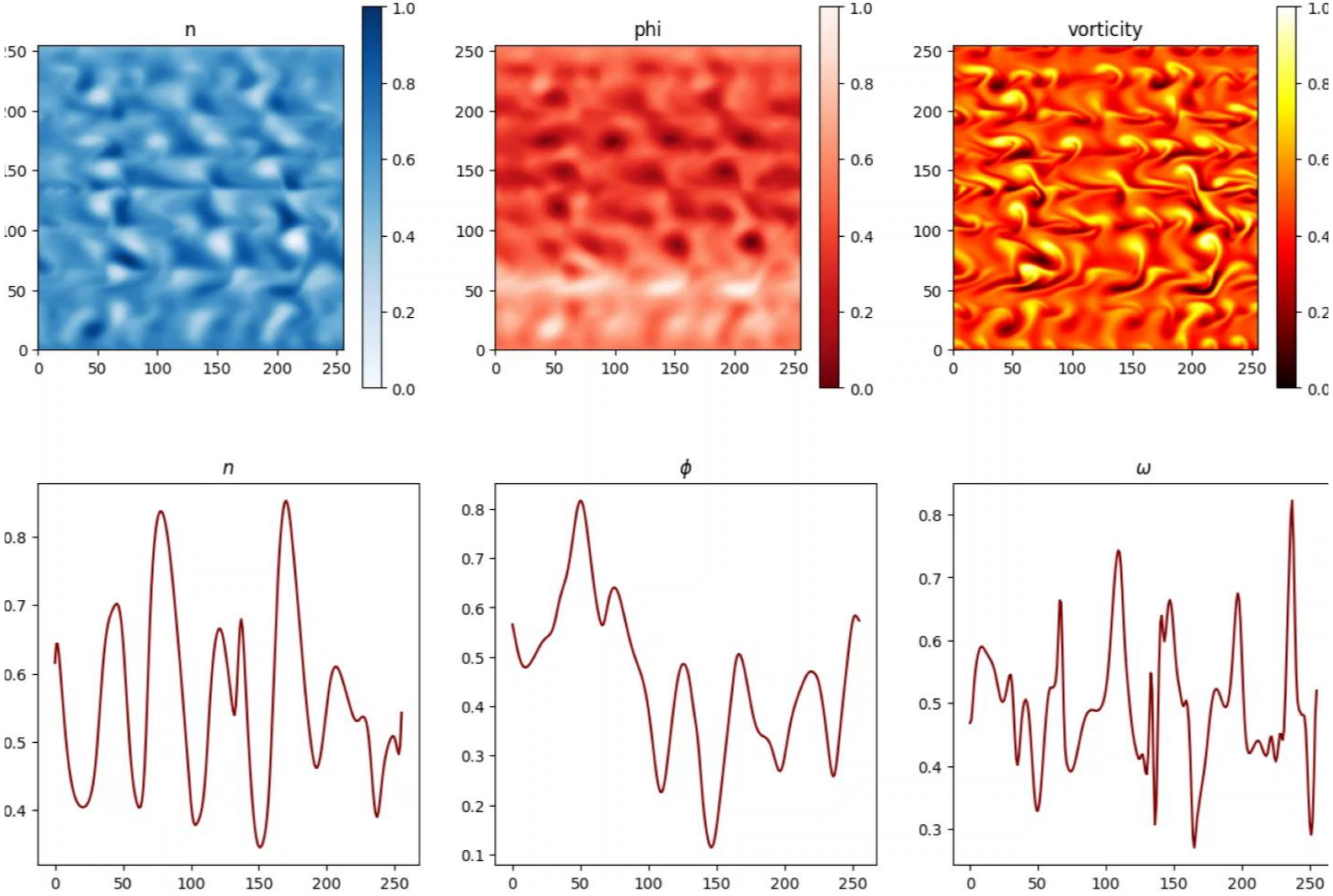
$$\frac{\partial n}{\partial t} + \{\phi, n\} = \alpha(\phi - n) - \kappa\frac{\partial \phi}{\partial y} - \mu\nabla^4 n$$

HW equations

Science and Technology Facilities Council

Hartree Centre

Modified Hasagawa Wakatani (mHW)



Results obtained with BOUT++

$$\text{zonal}: \langle f \rangle \equiv \frac{1}{L_y} \int f dy, \quad \text{nonzonal}: \tilde{f} \equiv f - \langle f \rangle$$

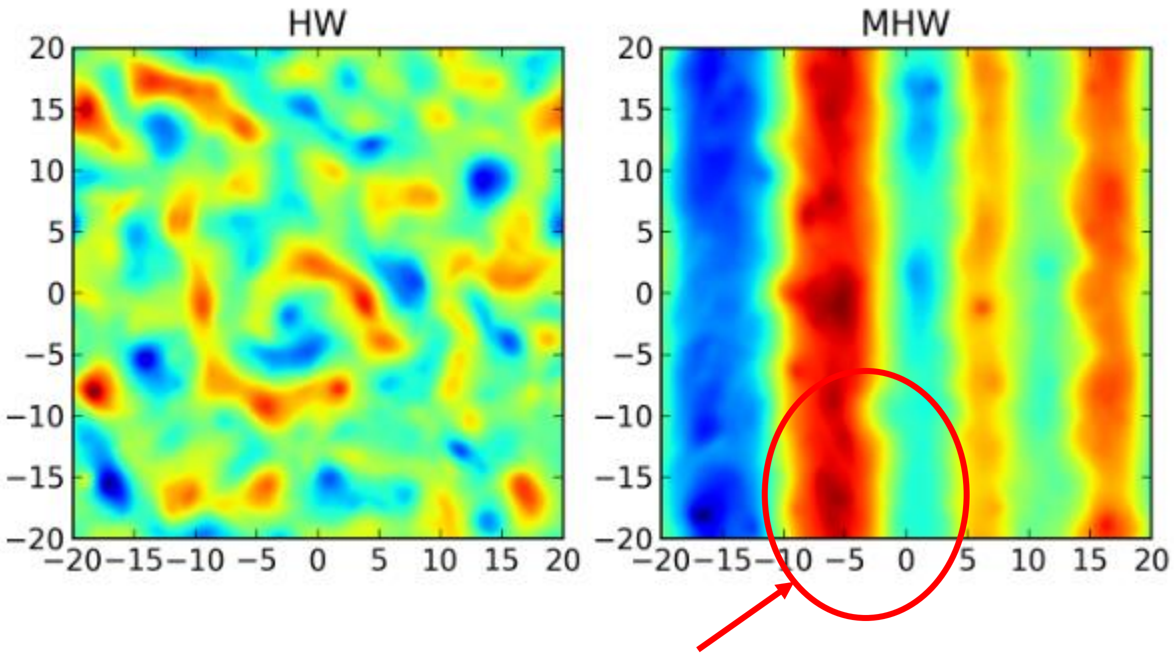
$$\frac{\partial \zeta}{\partial t} + \{\phi, \zeta\} = \alpha(\tilde{\phi} - \tilde{n}) - \mu \nabla^4 \zeta$$

$$\frac{\partial n}{\partial t} + \{\phi, n\} = \alpha(\tilde{\phi} - \tilde{n}) - \kappa \frac{\partial \phi}{\partial y} - \mu \nabla^4 n$$

In a tokamak edge any potential fluctuations on a flux surface is neutralized by the parallel electron motion

Numata, R., Ball, R., & Dewar, R. L, "Bifurcation in electrostatic resistive drift wave turbulence". *Physics of Plasmas*, **14** (10), 102312, 2007

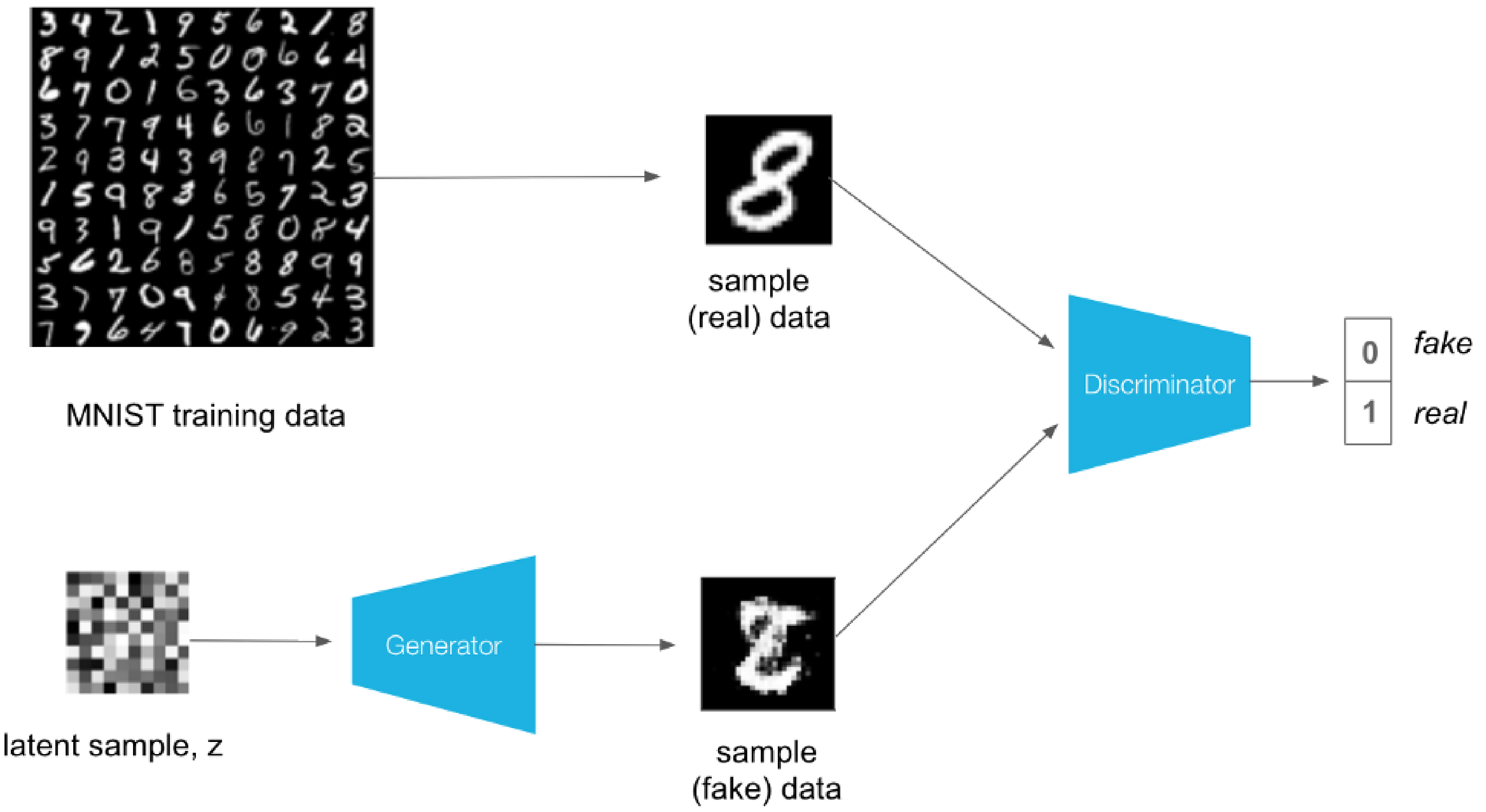
HW vs mHW



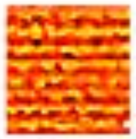
Zonal flow due to a gradient in the vertical direction

These are very difficult to model in LES!

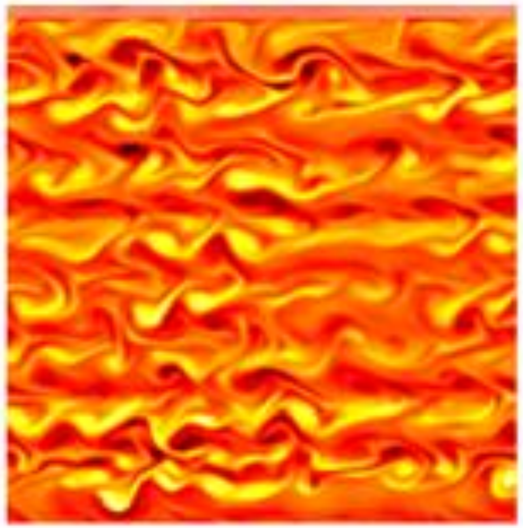
Generative Adversarial Networks (GANs)



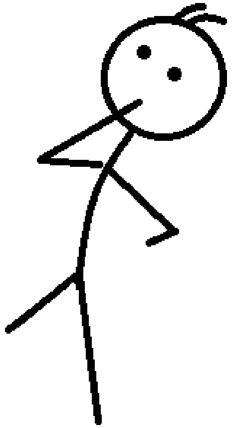
Idea: Can I train a GAN to reconstruct the DNS fields from the internal fields seen as LES fields?



LES field



DNS field

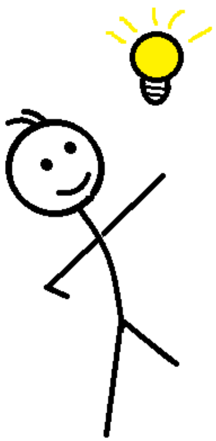


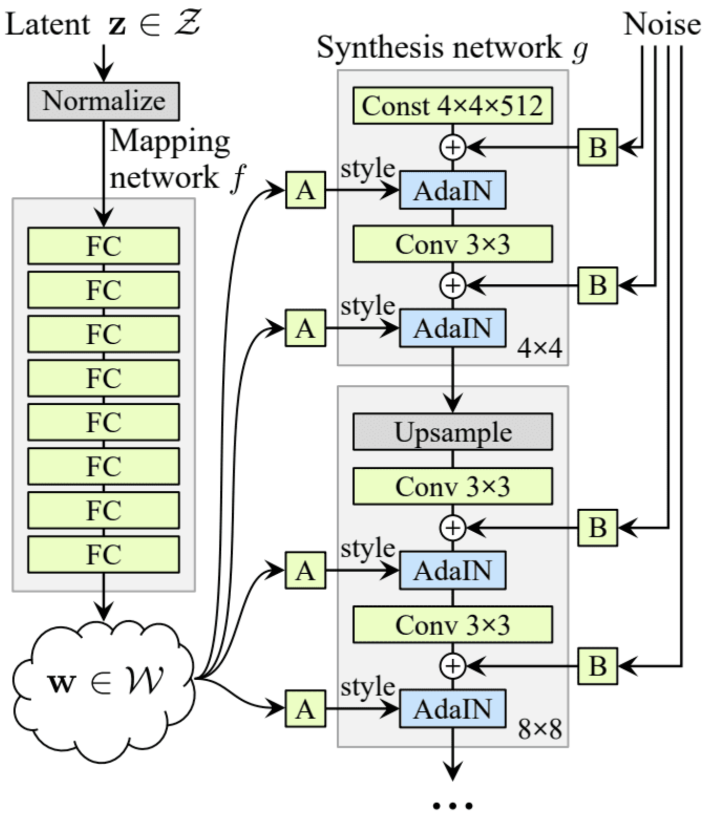
Potentially two instantaneous of the same Navier-Stoke problem can be obtained, **$U(t)$** and **$U(t+\Delta t)$** but there is no guarantee that the internal layers are representation of the same filtered Navier-Stoke problem, **$\tilde{U}(t)$** and **$\tilde{U}(t+\Delta t)$** !

Science and Technology Facilities Council

Hartree Centre

Idea: I need a more "flexible GAN": StyleGAN!





Our generator thinks of an image as a collection of "styles", where each style controls the effects at a particular scale

- Coarse styles ⟶ pose, hair, face shape
- Middle styles ⟶ facial features, eyes
- Fine styles ⟶ color scheme

Each layer (style) can be adjusted without interfering with the other levels!

Latent space interpolation

Coarse
Styles ($4^2 - 8^2$)

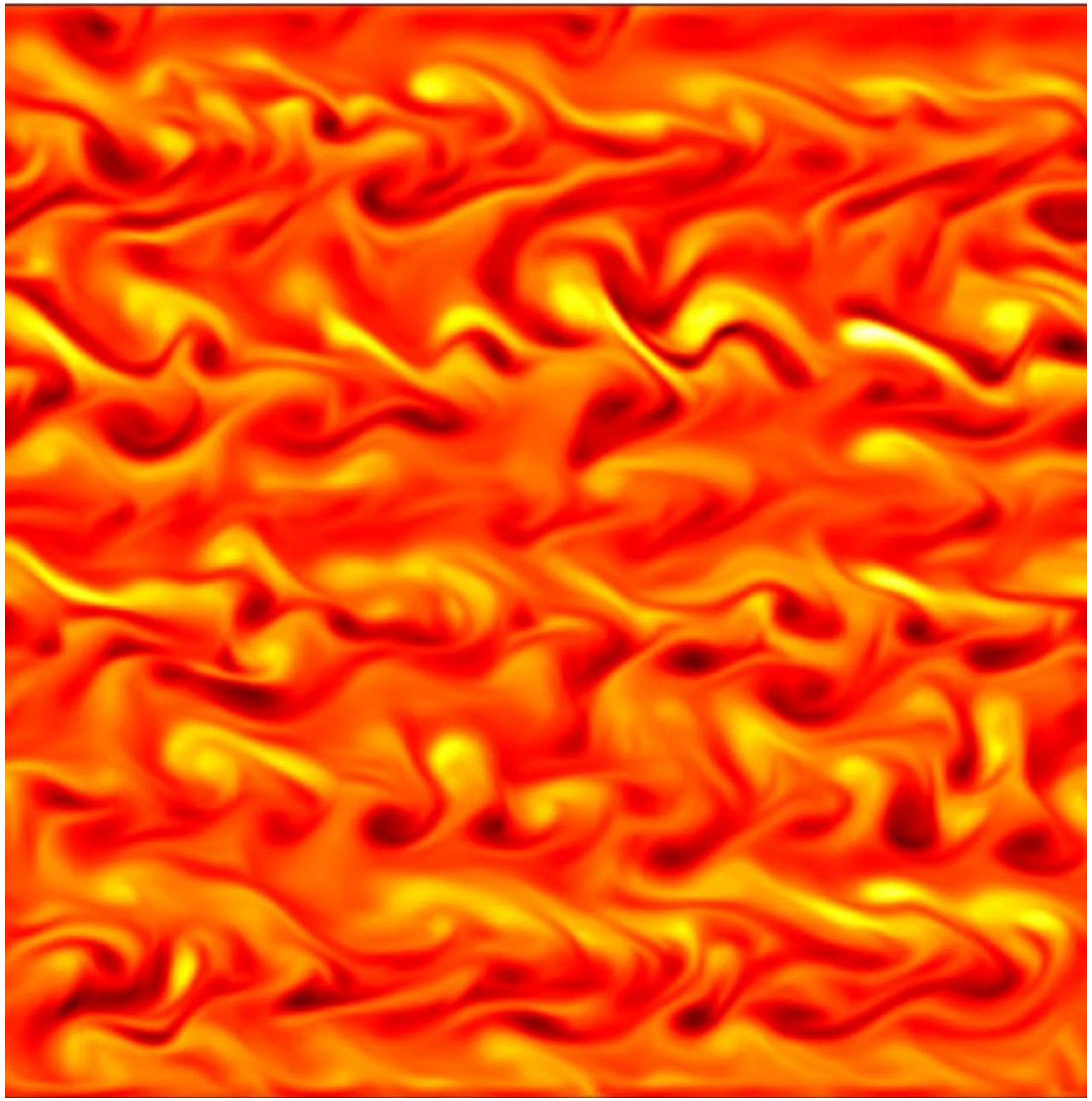
W_a W_b

Medium
Styles ($16^2 - 32^2$)

W_a W_b

Fine
Styles ($64^2 - 512^2$)

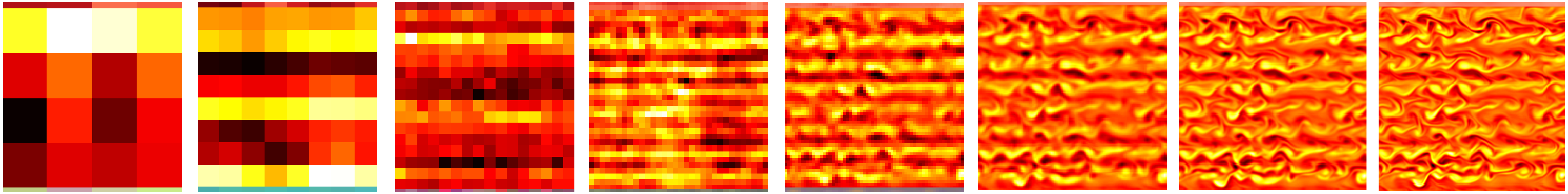
W_a W_b



Science and Technology Facilities Council
Hartree Centre

How StyleGAN is linked to LES?

Different layers of the StylES generator



4x4 8x8 16x16 32x32 64x64 128x128 256x256 512x512

Different layer can be "thought" as different filtered LES fields!

We can use StyleGAN for deconvolution of a LES field and find corresponding DNS field

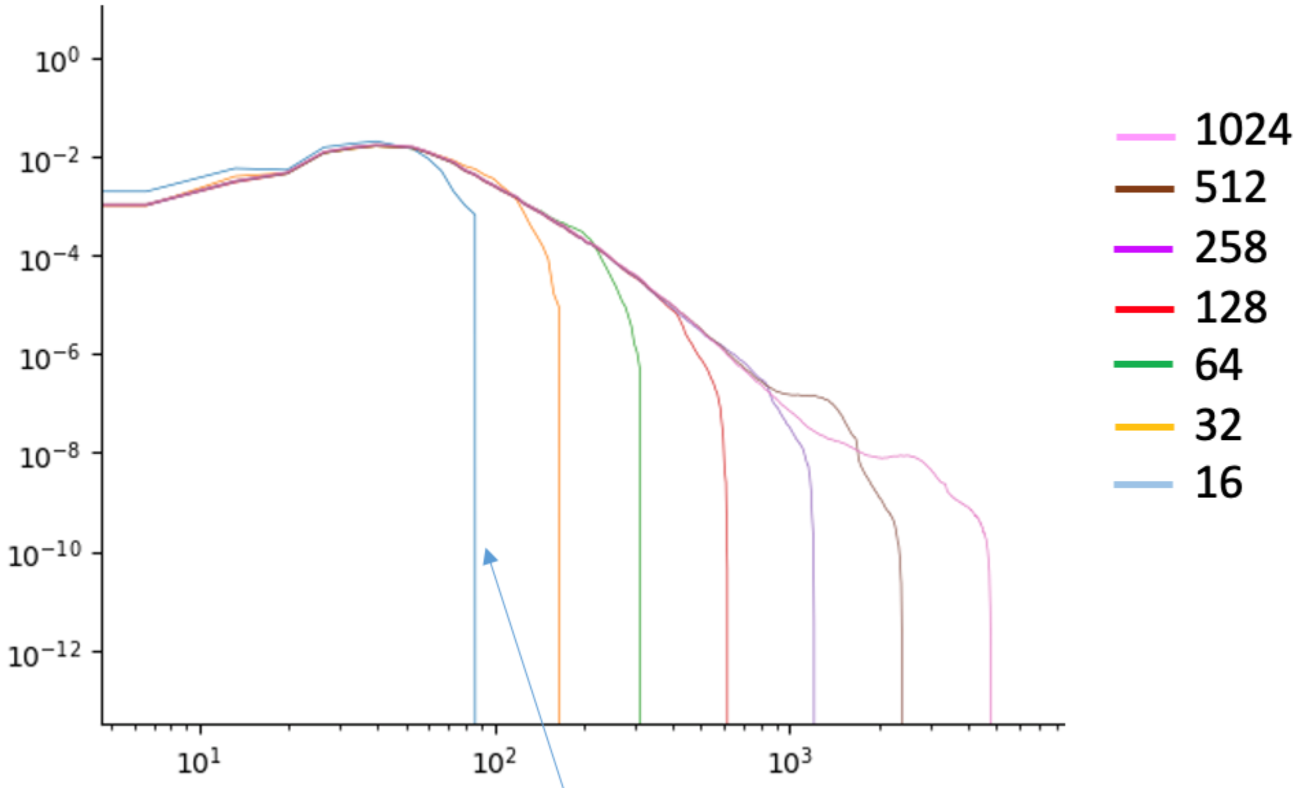
We do not need a RNN!

Science and Technology Facilities Council

Hartree Centre

Spectra at different layers of StyleGAN (1024^2)

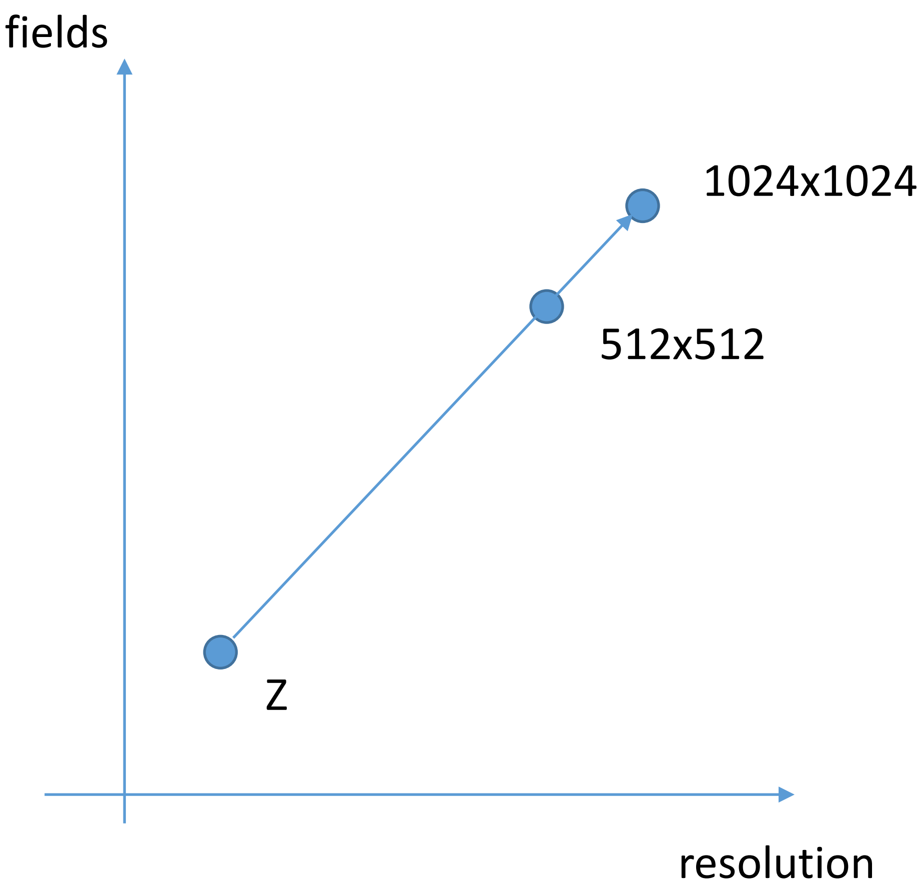
We want StyleGAN learn in the universal equilibrium range!

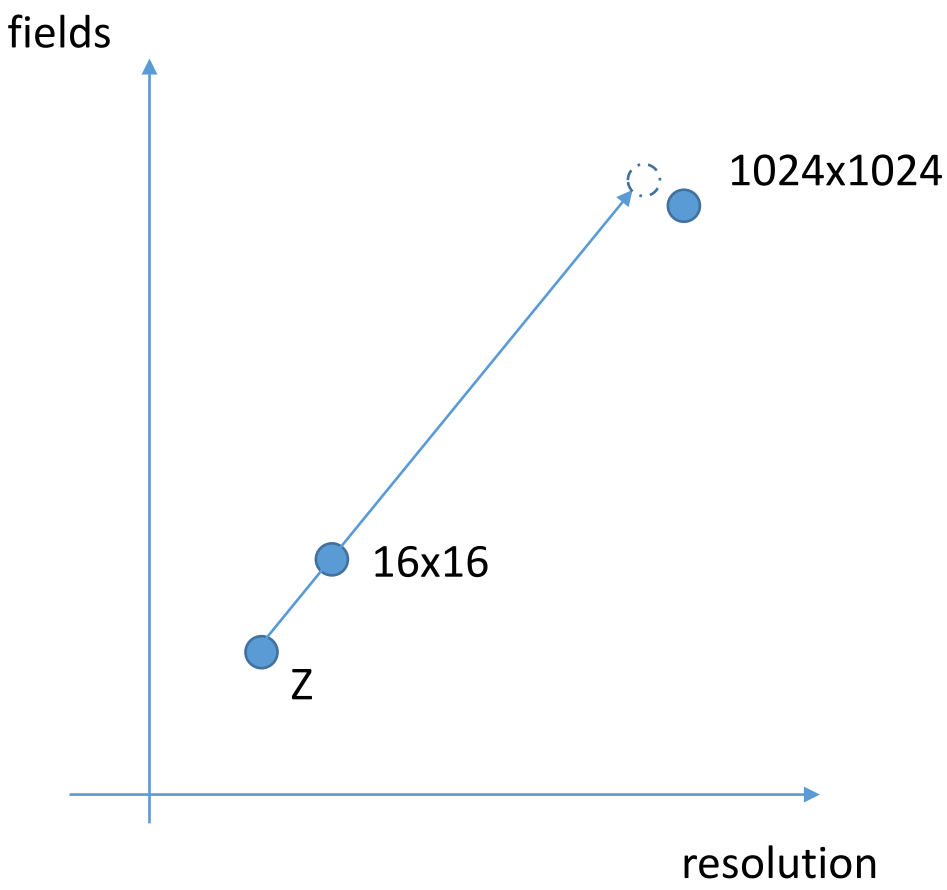


We can choose any layer down to 16x16!

Hartree Centre

Pros and cons





fast research in latent space (find Z) -> slow LES

slow research in latent space -> fast LES

Style Eddy Simulation (StylES)

Procedure:

1) generate DNS data

2) train StyleGAN on the data together with a filter *G* from DNS to LES fields

3) pick a style (LES field) within the universal equilibrium range

4) start from a given DNS field => find latent space W^+ modifying each style up to the LES internal layer

5) find non filtered linear term $\overline{UU}$ from the reconstructed DNS field and filter *G*

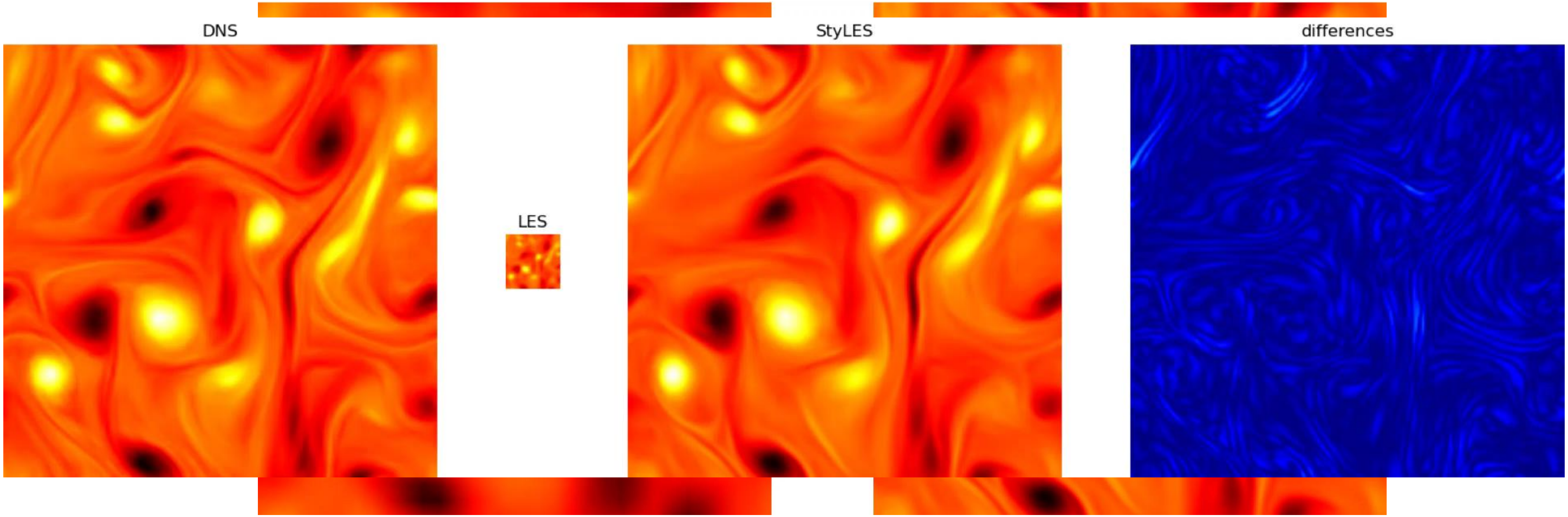
6) move in time using LES equations from *t* to *t+Δt*

7) repeat from step 3, matching the new LES field at time *t+Δt*

Science and Technology Facilities Council
Hartree Centre

Results on 2D-HIT

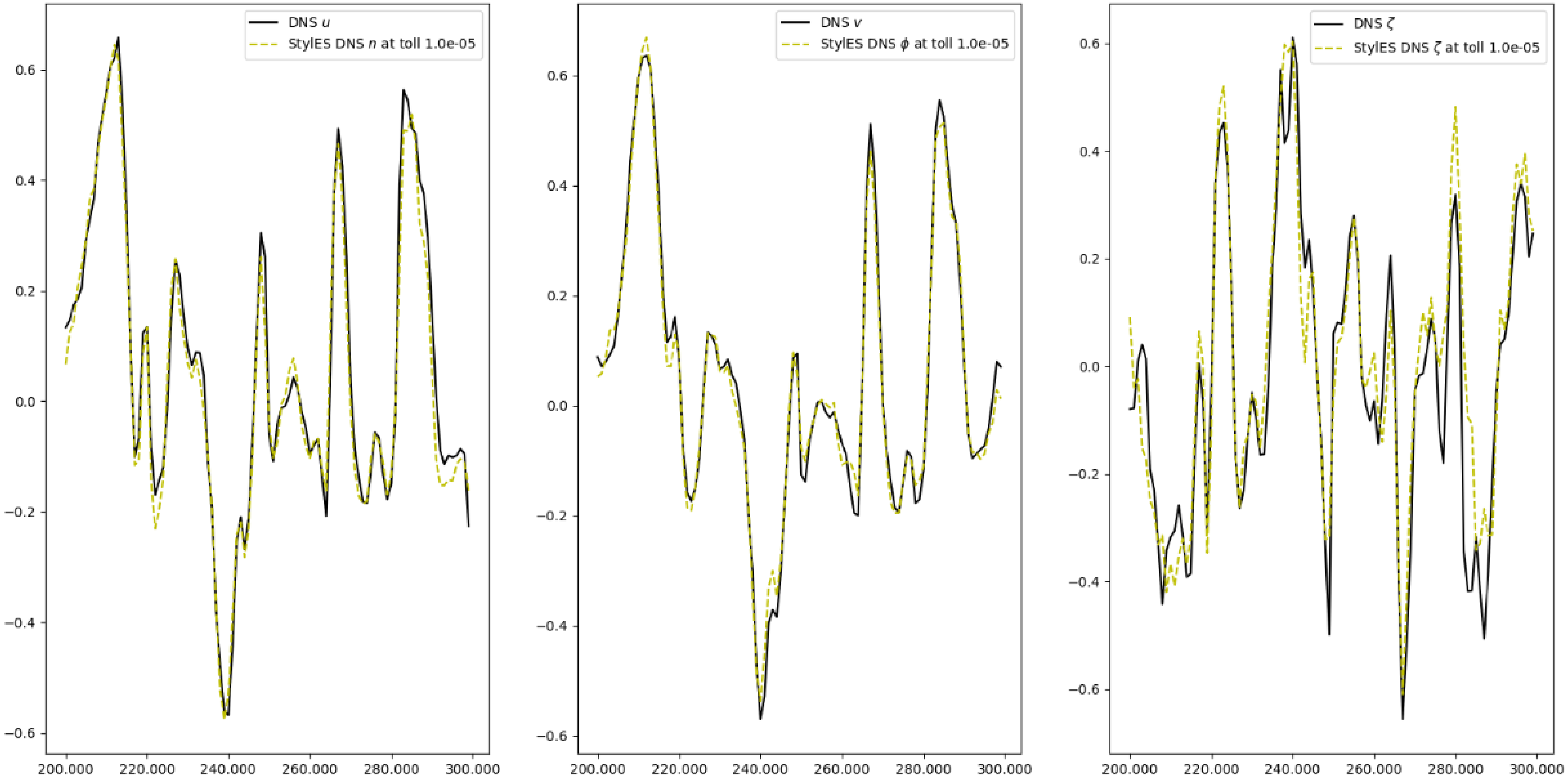
StylES 32x32 StylES 256x256



Reconstruction from 32x32 -> 256x256, $\varepsilon_{REC} = 10^{-4}$

Science and Technology Facilities Council
Hartree Centre

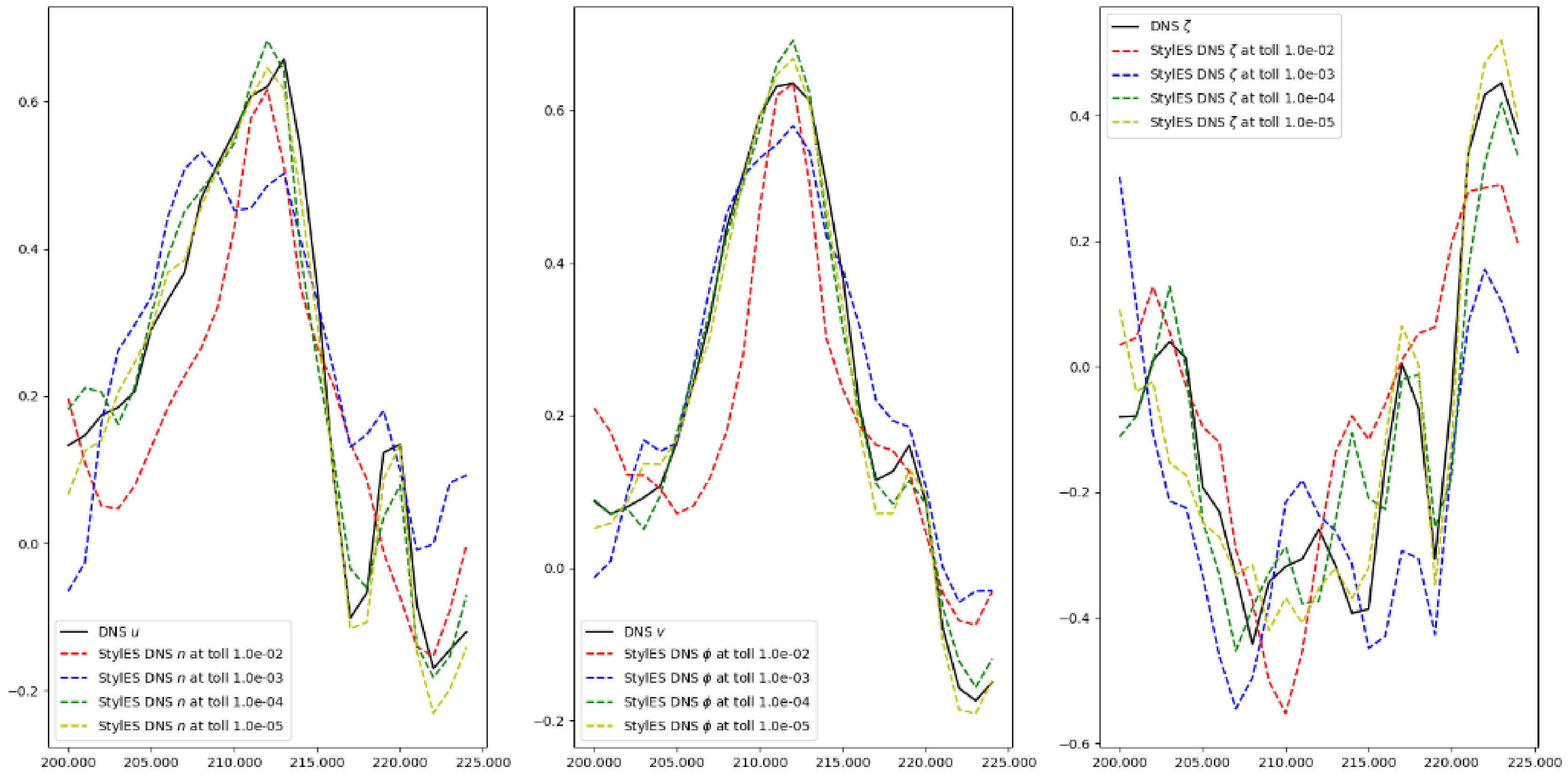
Results on HW (I)



Reconstruction across the full training range (200 to 300 ω_i)

Science and Technology Facilities Council
Hartree Centre

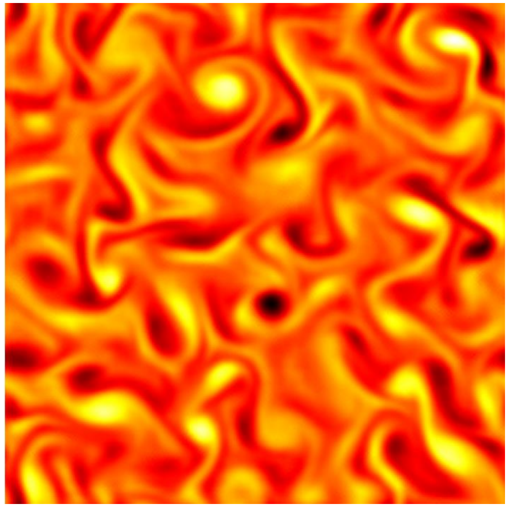
Results on HW (II)



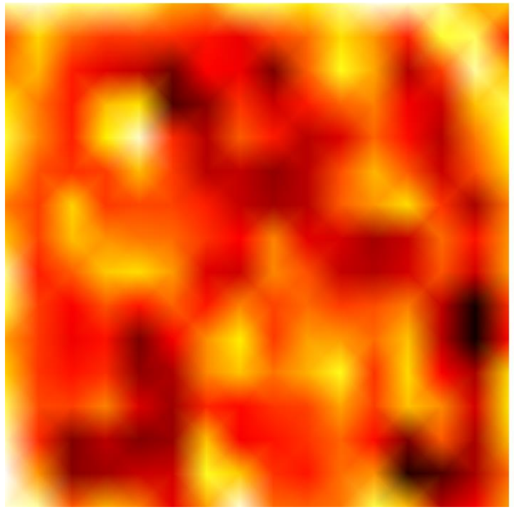
Convergence to DNS results as we tight the reconstruction tolerance ε_{REC}

Science and Technology Facilities Council
Hartree Centre

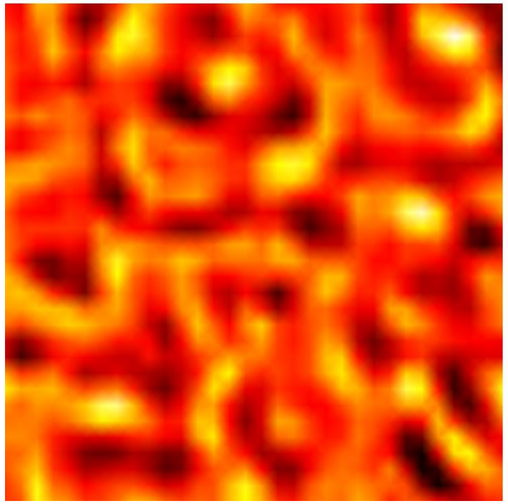
HW field reconstruction



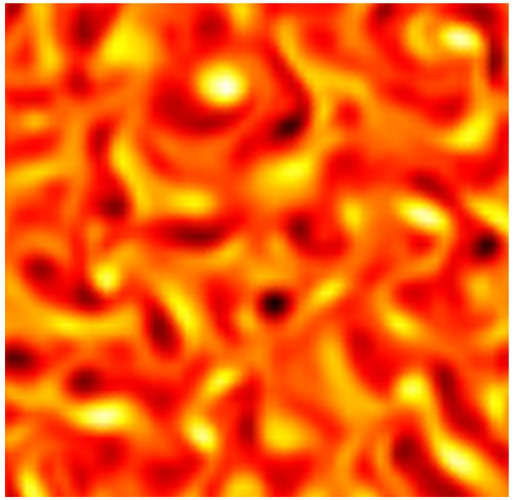
DNS (256x256)



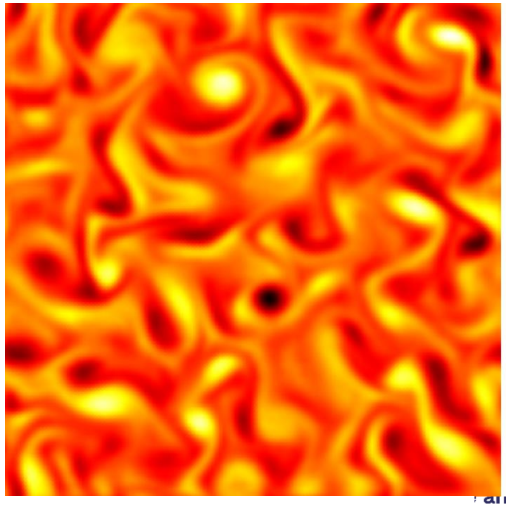
16x16



32x32

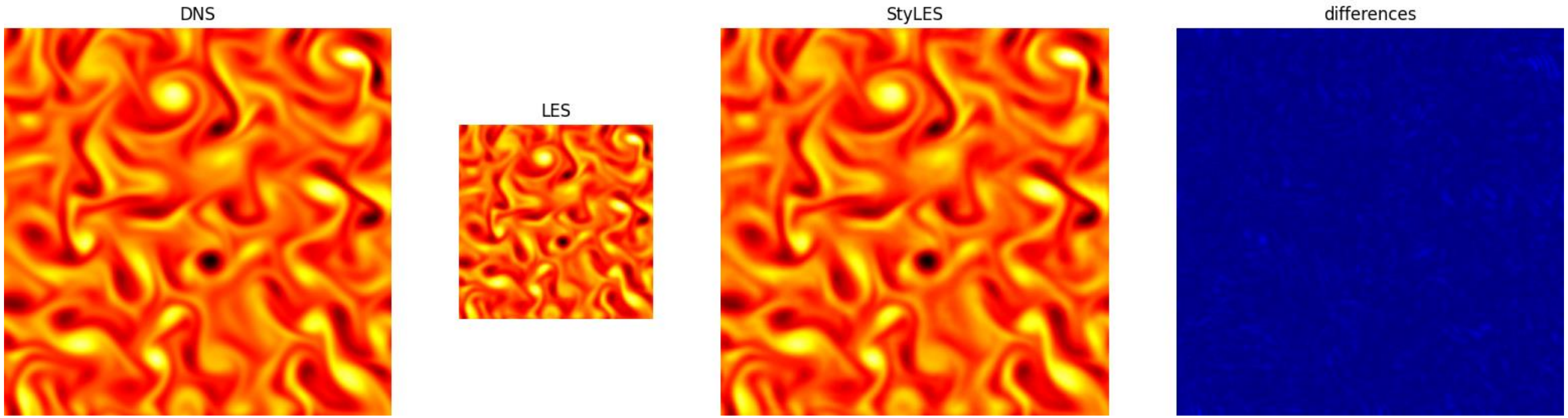


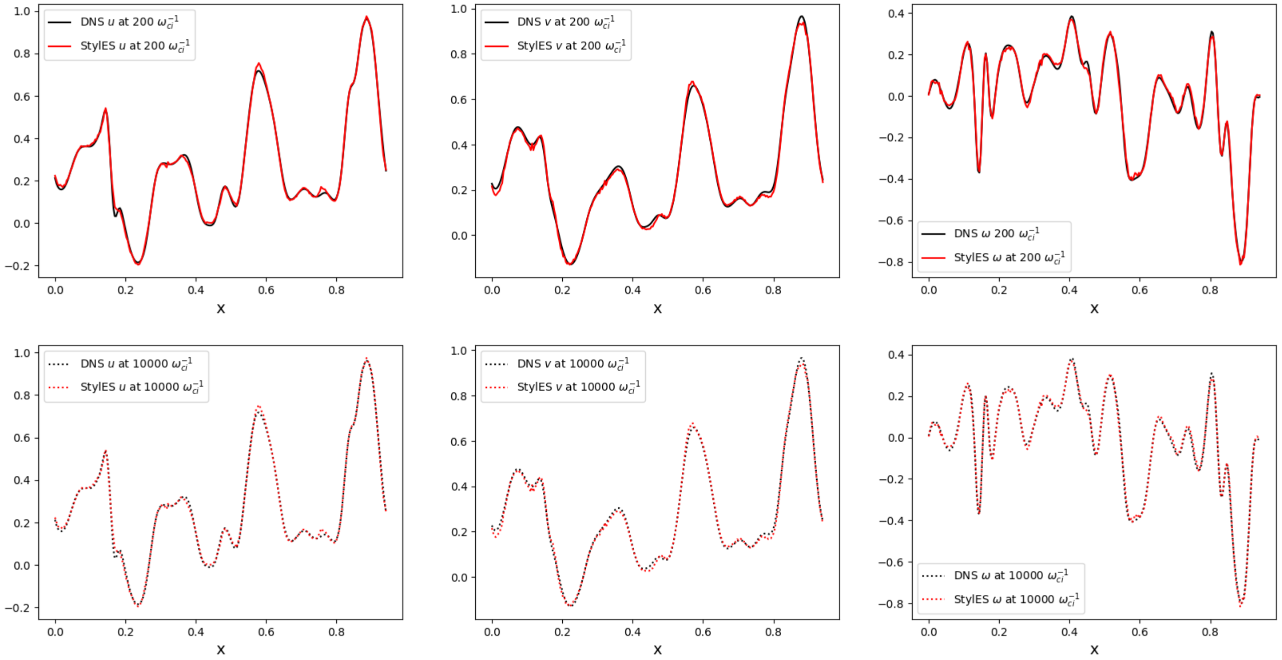
64x64

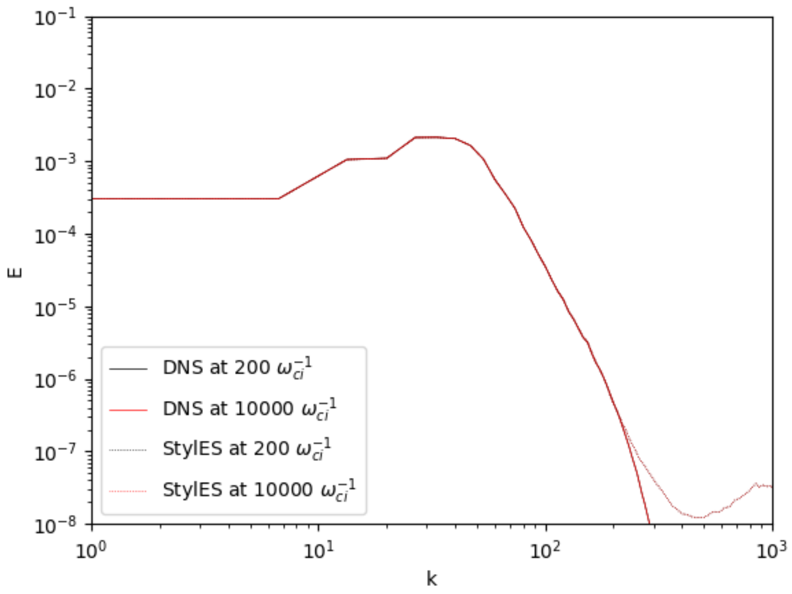


128x128

HW 128x128 (toll $2x10^{-4}$)



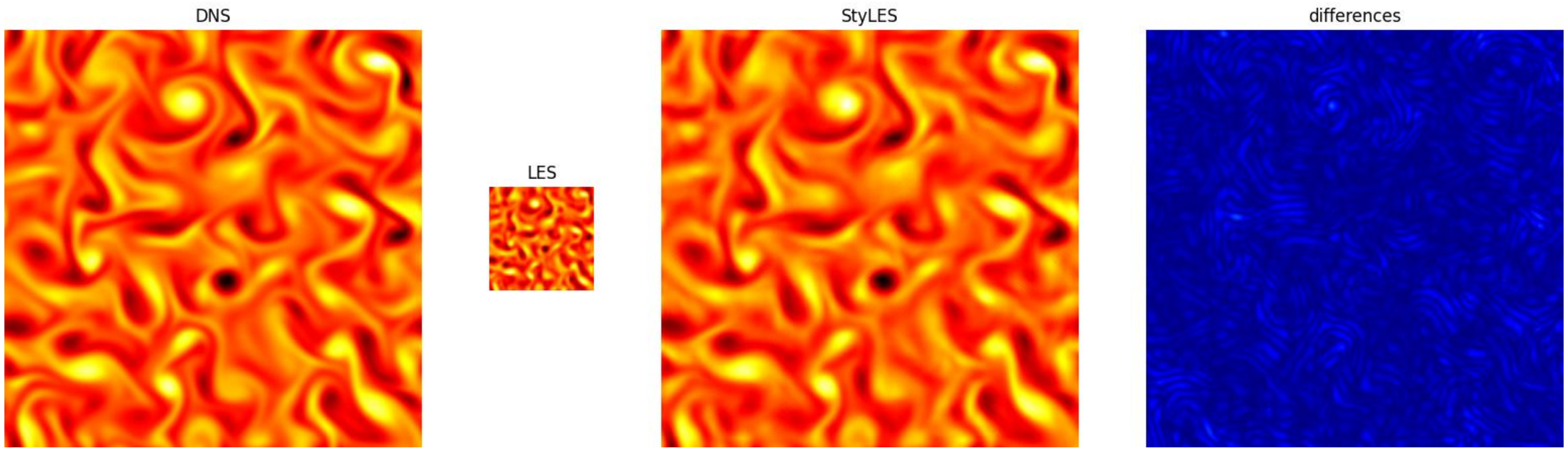


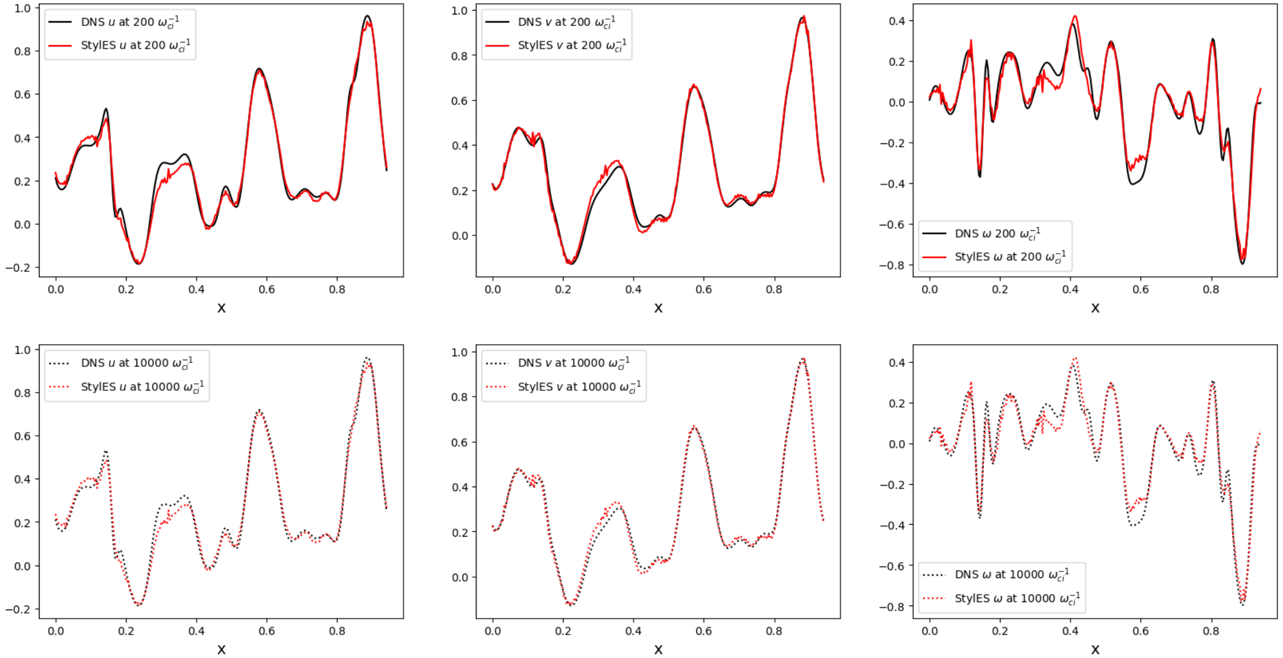


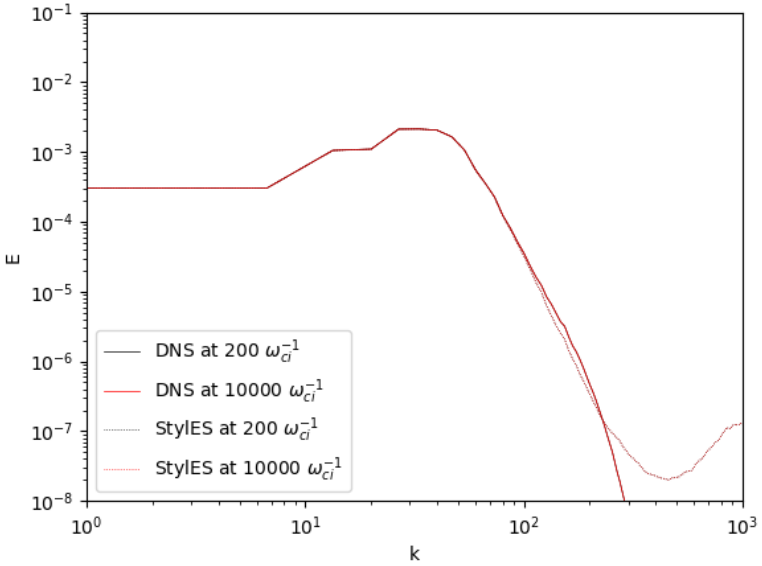
Science and Technology Facilities Council

Hartree Centre

HW 64x64 (toll $2x10^{-4}$)

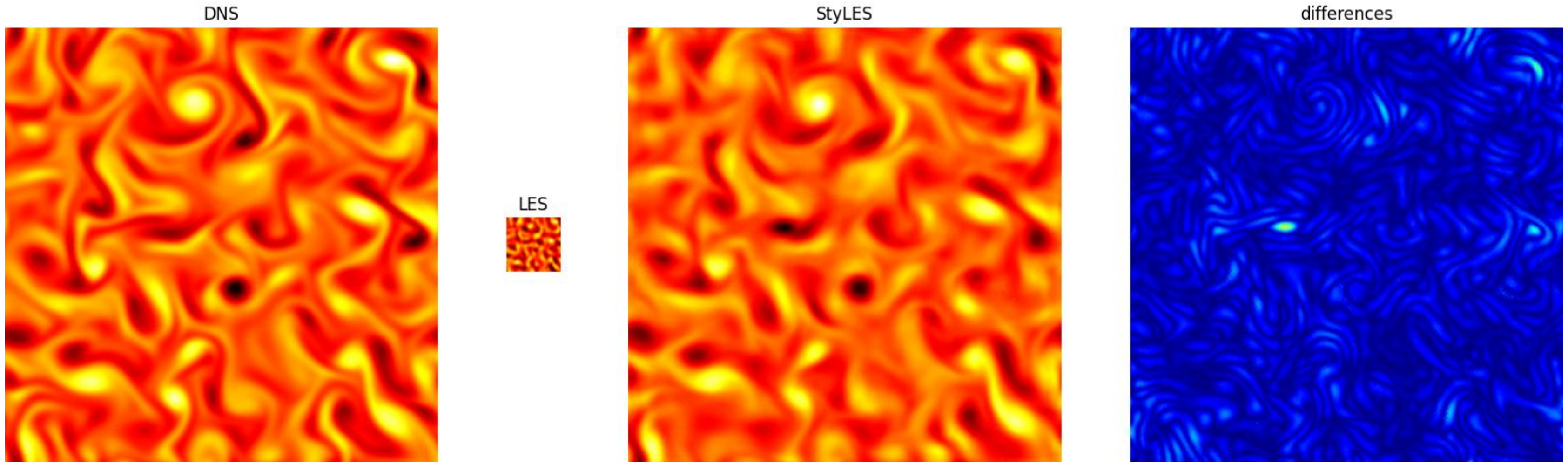


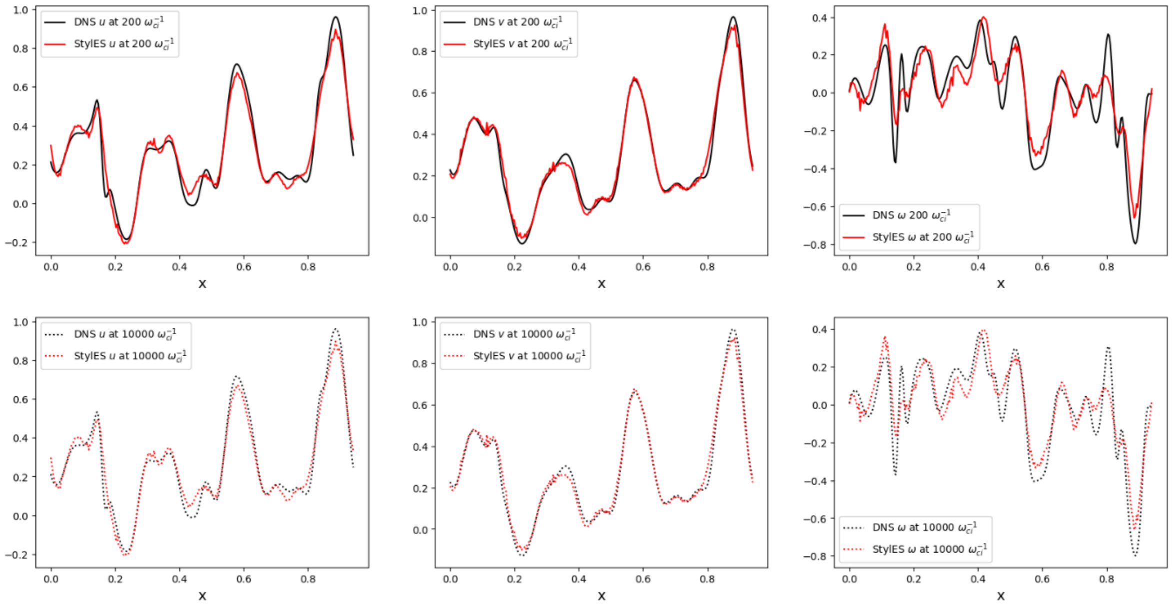


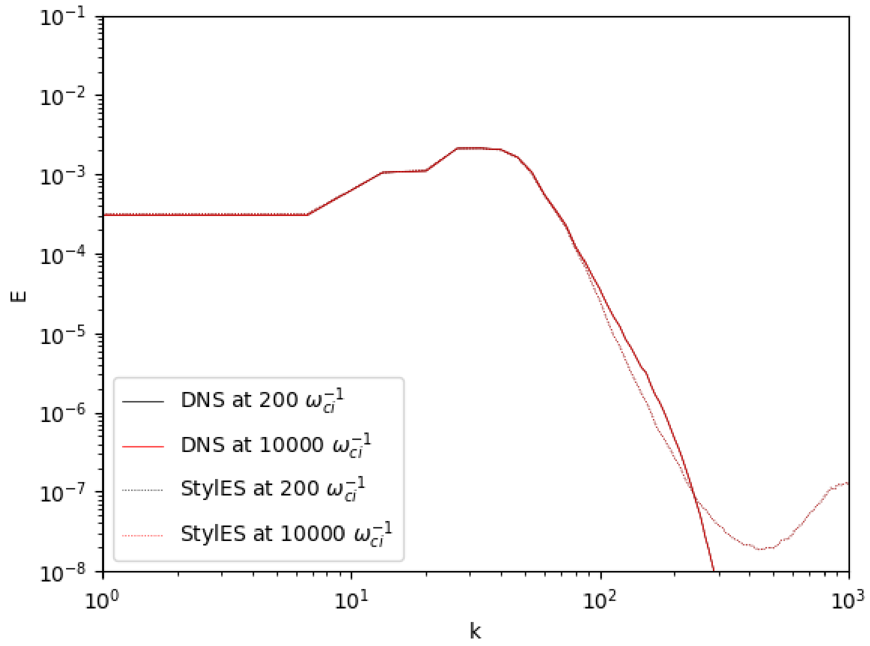


Hartree Centre

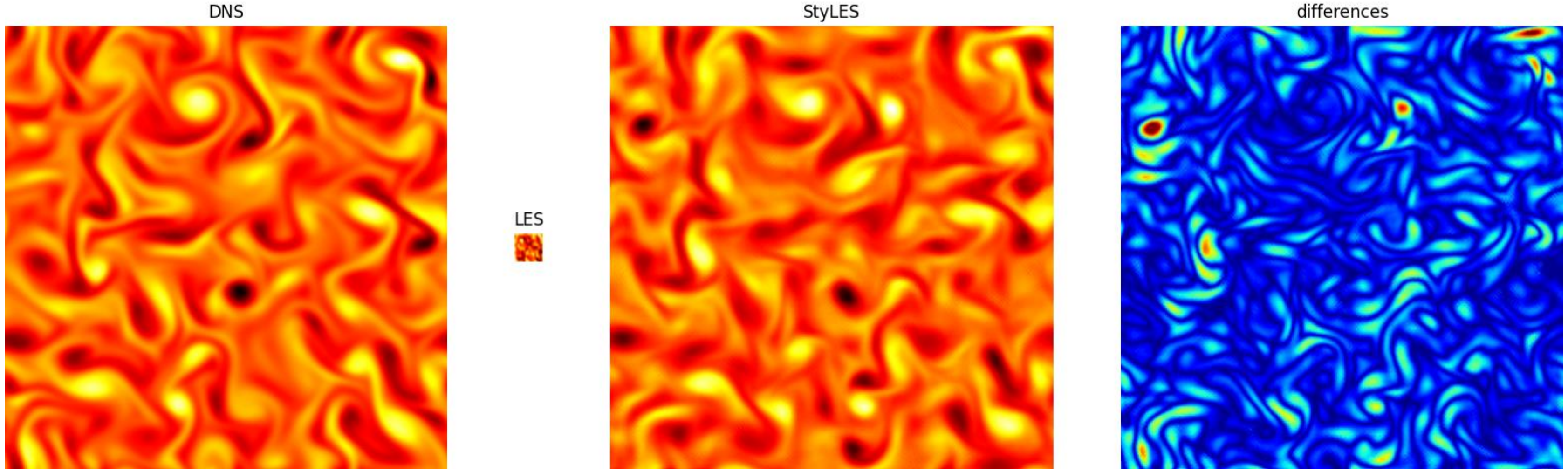
HW 32x32 (toll $5x10^{-5}$)

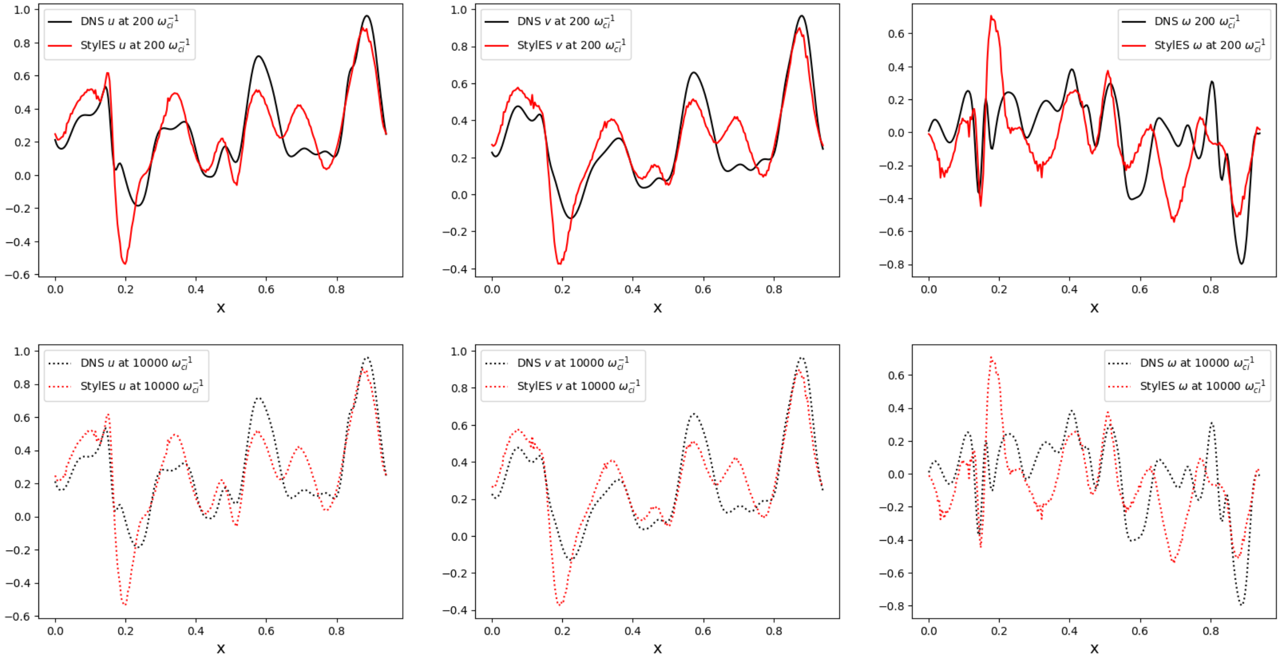


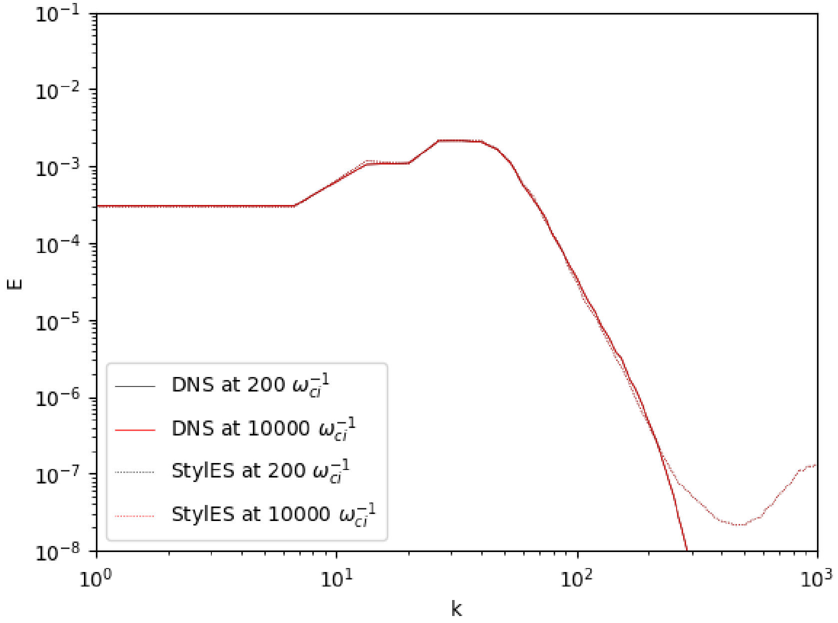




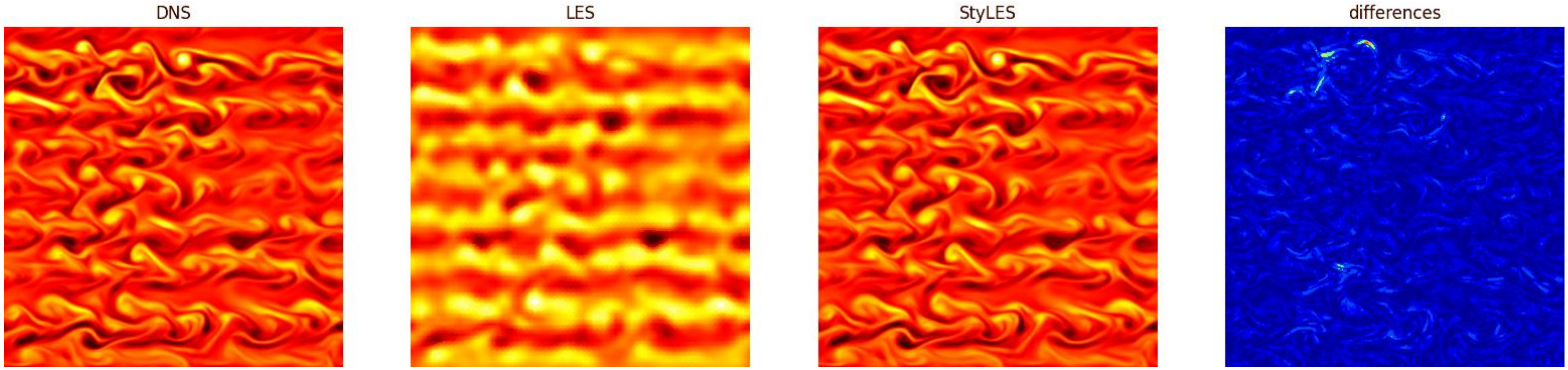
HW 16x16 (toll 10^{-6})







Results on mHW (II)



512x512 | 64x64 | 512x512 | 512x512

Reconstruction with tolerance 10^{-5}

Integration with BOUT++

Filtered form of HW equations

$$\frac{\partial\tilde{\zeta}}{\partial t}+\frac{\partial\tilde{\phi}}{\partial y}\frac{\partial\tilde{\zeta}}{\partial x}-\frac{\partial\tilde{\phi}}{\partial x}\frac{\partial\tilde{\zeta}}{\partial y}=\alpha(\tilde{\phi}-\tilde{n})-\mu_\omega\nabla^4\tilde{\zeta}+D_{\phi_y\zeta_x}+D_{\phi_x\zeta_y}$$

$$\frac{\partial\tilde{n}}{\partial t}+\frac{\partial\tilde{\phi}}{\partial y}\frac{\partial\tilde{n}}{\partial x}-\frac{\partial\tilde{\phi}}{\partial x}\frac{\partial\tilde{n}}{\partial y}=\alpha(\tilde{\phi}-\tilde{n})-k\frac{\partial\tilde{\phi}}{\partial y}-\mu_n\nabla^4\tilde{n}+D_{\phi_y n_x}+D_{\phi_x n_y}$$

$\tilde{n}\quad\tilde{\phi}\quad\tilde{\zeta}$

are the LES fields to be passed to StylEGAN running on GPU via TensorFlow

where:

$$\widetilde{\frac{\partial\phi}{\partial y}\frac{\partial\zeta}{\partial x}}-\frac{\partial\tilde{\phi}}{\partial y}\frac{\partial\tilde{\zeta}}{\partial x}=D_{\phi_y\zeta_x}$$

$$\widetilde{\frac{\partial\phi}{\partial x}\frac{\partial\zeta}{\partial y}}-\frac{\partial\tilde{\phi}}{\partial x}\frac{\partial\tilde{\zeta}}{\partial y}=D_{\phi_x\zeta_y}$$

$$\widetilde{\frac{\partial\phi}{\partial y}\frac{\partial n}{\partial x}}-\frac{\partial\tilde{\phi}}{\partial y}\frac{\partial\tilde{n}}{\partial x}=D_{\phi_y n_x}$$

$$\widetilde{\frac{\partial\phi}{\partial x}\frac{\partial n}{\partial y}}-\frac{\partial\tilde{\phi}}{\partial x}\frac{\partial\tilde{n}}{\partial y}=D_{\phi_x n_y}$$

LES size fields to be passed back to BOUT++

Integration with BOUT++

```
rLES = findLESTerms(n, phi, vort, pModule, pFindLESTerms);
int N_LES = n.getNz();
int cont=0;
for(int i=2; i<n.getNx()-2; i++)   // we assume 2 guards cells in x-direction
  for(int j=0; j<1; j++)
    for(int k=0; k<n.getNz(); k++){
      Dpyvx(i,j,k) = rLES[cont + 0*N_LES*N_LES];
      Dpxvy(i,j,k) = rLES[cont + 1*N_LES*N_LES];
      Dpynx(i,j,k) = rLES[cont + 2*N_LES*N_LES];
      Dpxny(i,j,k) = rLES[cont + 3*N_LES*N_LES];
      cont = cont+1;
    }

ddt(n) = -Dn*Delp4(n) + Dpyvx + Dpxvy;
ddt(vort) = -Dvort*Delp4(vort) + Dpynx + Dpxny;
```

call a function with an Embedded Python call

pass back 1D numpy array to BOUT++

add sub-grid scale terms

hw.cxx file in Hasegawa-wakatani example

https://github.com/farscape-project/BOUT-dev.git

branch: bout_with_StylES

What happens on the GPU?

$\tilde{n} \quad \tilde{\phi} \quad \tilde{\zeta}$

unpack 1D numpy array from BOUT++ (CPU, 64bit)

$\tilde{n} \quad \tilde{\phi} \quad \tilde{\zeta}$

TensorFlow arrays (GPU, 32bit)

$n \quad \phi \quad \zeta$

reconstruct DNS fields via search in the latent space (GPU)

$$\frac{\partial n}{\partial x}, \frac{\partial n}{\partial y}, \frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y}, \frac{\partial \zeta}{\partial x}, \frac{\partial \zeta}{\partial y}$$

find derivatives 2nd order central scheme (GPU)

$$\frac{\partial \phi}{\partial y}\frac{\partial \zeta}{\partial x}, \frac{\partial \phi}{\partial x}\frac{\partial \zeta}{\partial y}, \frac{\partial \phi}{\partial y}\frac{\partial n}{\partial x}, \frac{\partial \phi}{\partial x}\frac{\partial n}{\partial y}$$

product as derivatives (GPU)

$$\widetilde{\frac{\partial \phi}{\partial y}\frac{\partial \zeta}{\partial x}} \quad \widetilde{\frac{\partial \phi}{\partial x}\frac{\partial \zeta}{\partial y}} \quad \widetilde{\frac{\partial \phi}{\partial y}\frac{\partial n}{\partial x}} \quad \widetilde{\frac{\partial \phi}{\partial x}\frac{\partial n}{\partial y}}$$

$$\widetilde{\frac{\partial \phi}{\partial y}} \quad \widetilde{\frac{\partial \zeta}{\partial x}} \quad \widetilde{\frac{\partial \phi}{\partial x}} \quad \widetilde{\frac{\partial \zeta}{\partial y}} \quad \widetilde{\frac{\partial \phi}{\partial y}} \quad \widetilde{\frac{\partial n}{\partial x}} \quad \widetilde{\frac{\partial \phi}{\partial x}} \quad \widetilde{\frac{\partial n}{\partial y}}$$

filter all quantities (GPU)

$$\widetilde{\frac{\partial \phi}{\partial y}\frac{\partial \zeta}{\partial x}} - \widetilde{\frac{\partial \phi}{\partial y}}\,\widetilde{\frac{\partial \zeta}{\partial x}} = D_{\phi_y \zeta_x}$$

...

find sub-grid scale terms (GPU)

$$D_{\phi_y \zeta_x}, D_{\phi_x \zeta_y}, D_{\phi_y n_x}, D_{\phi_x n_y}$$

pack to a 1D numpy array for BOUT++ (CPU, cast to 64bit)

Science and Technology Facilities Council

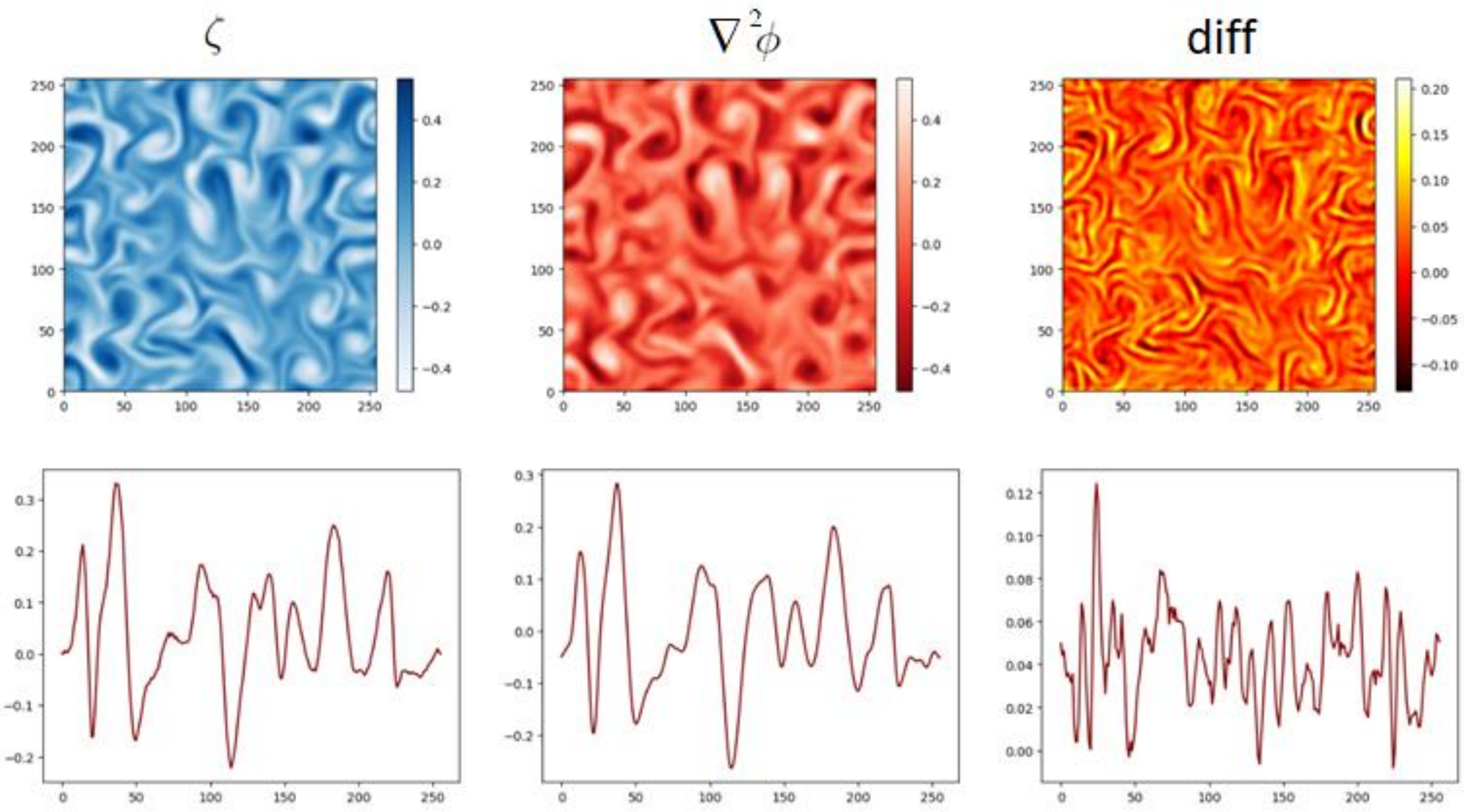
Hartree Centre

Current Issues (I)

$$\nabla_{\perp}^{2}\phi = \zeta$$

A well trained GAN should produce fields where this correlation is perfectly satisfied!

But this is not always the case…



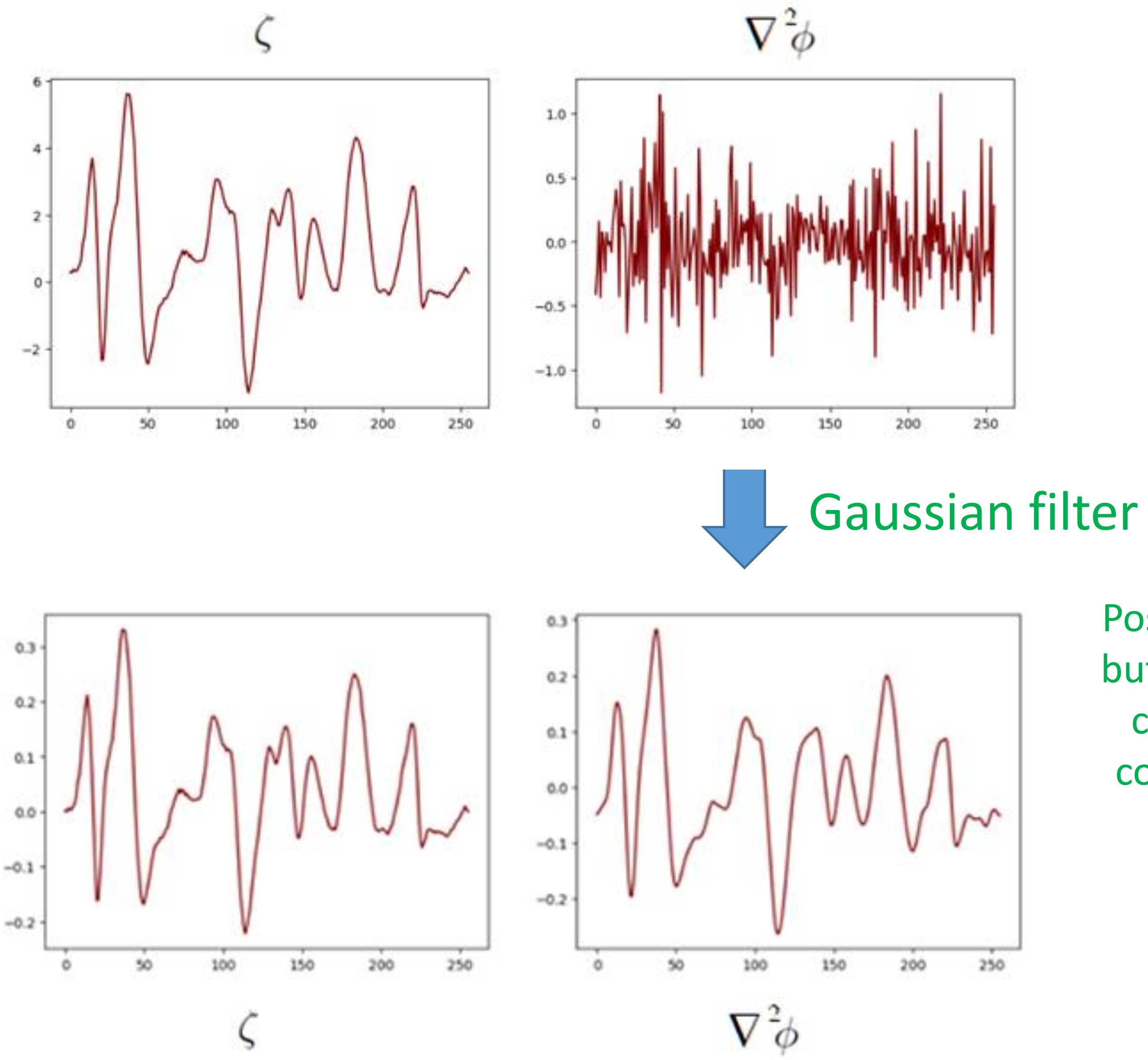
Science and Technology Facilities Council
Hartree Centre

Current Issues (II)

$$\nabla^2_\perp \phi = \zeta$$

oscillations may occur in the second derivatives field

(Maybe due to the image noise injected in the last style?)



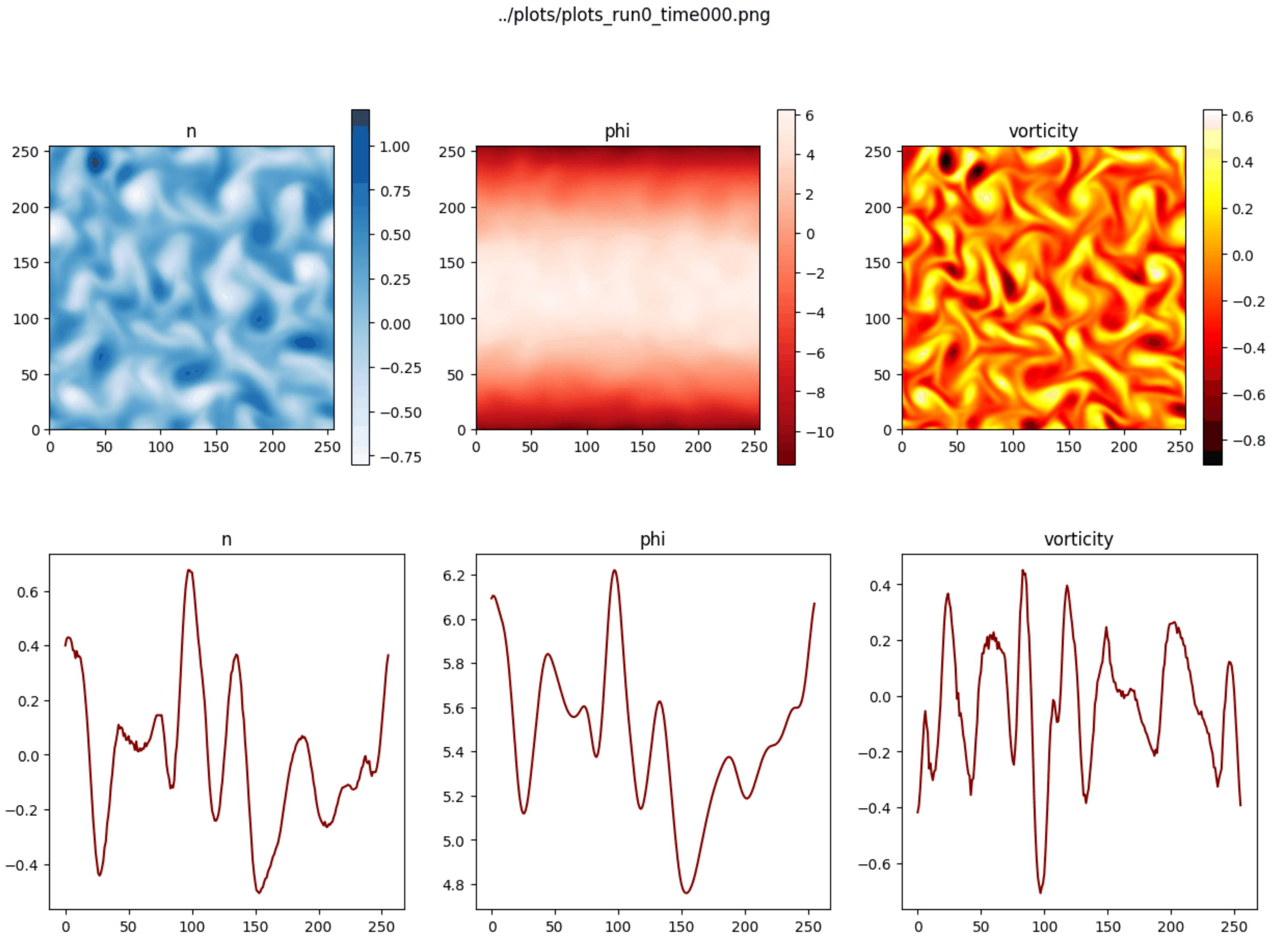
Possible solution, but it has an extra computational cost scaling with N^2!

UKRI Science and Technology Facilities Council
Hartree Centre

Current Issues (III)

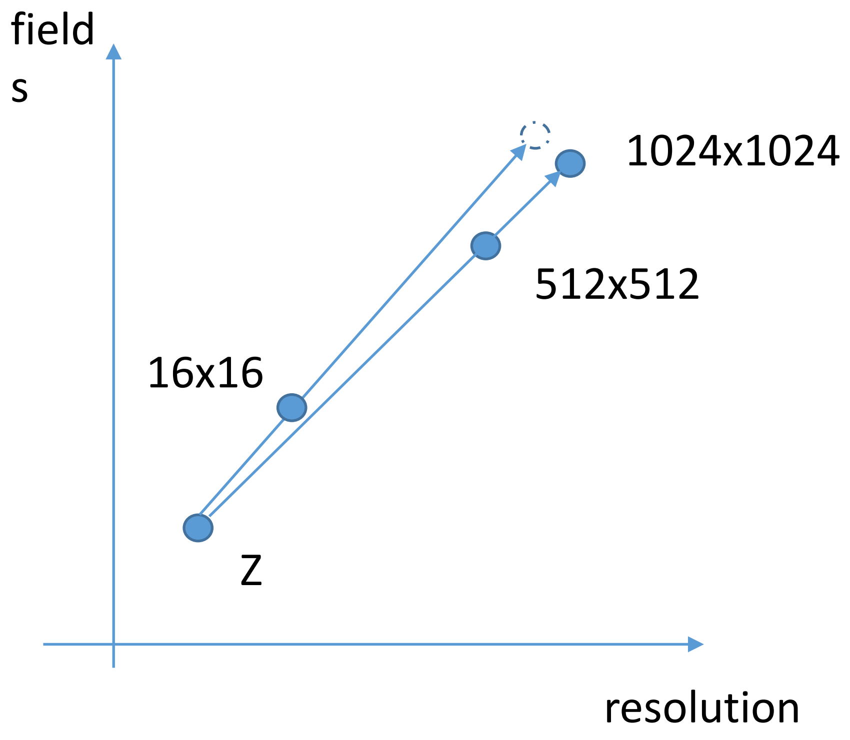
Net flow when starting BOUT++ from a StyleGAN DNS

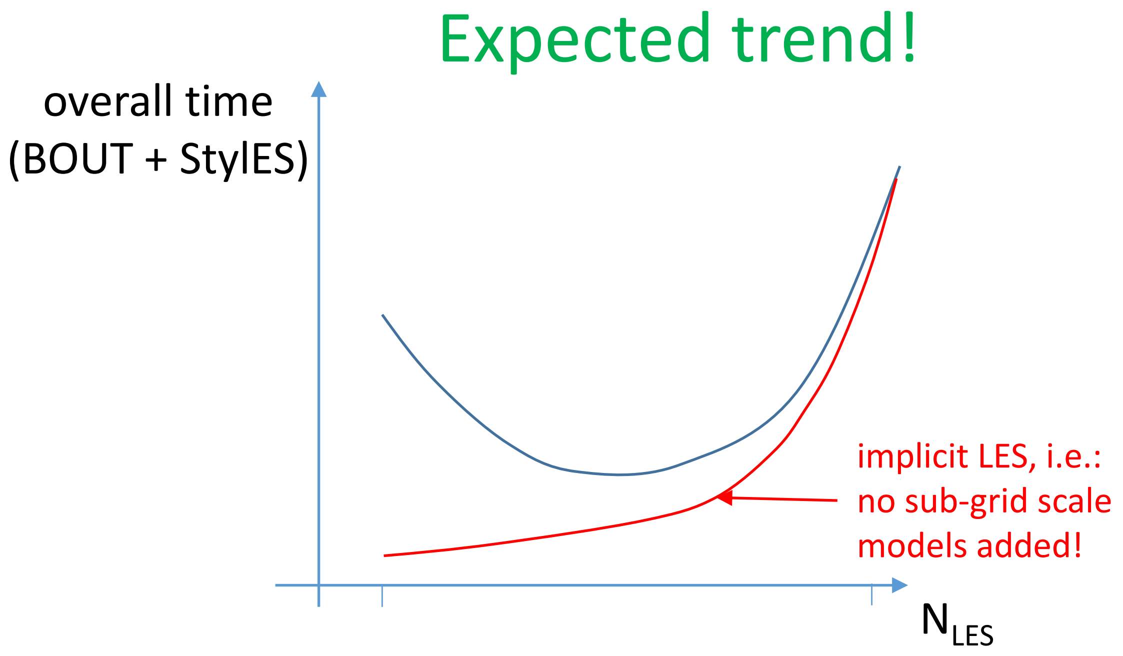
Enforce symmetry!



Science and Technology Facilities Council
Hartree Centre

Performance (I)





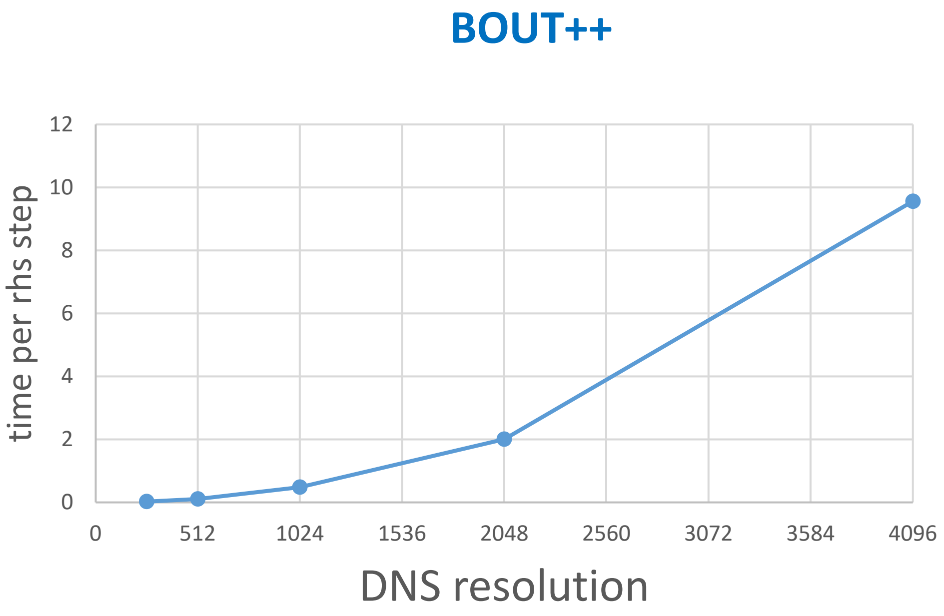
A smaller LES has:

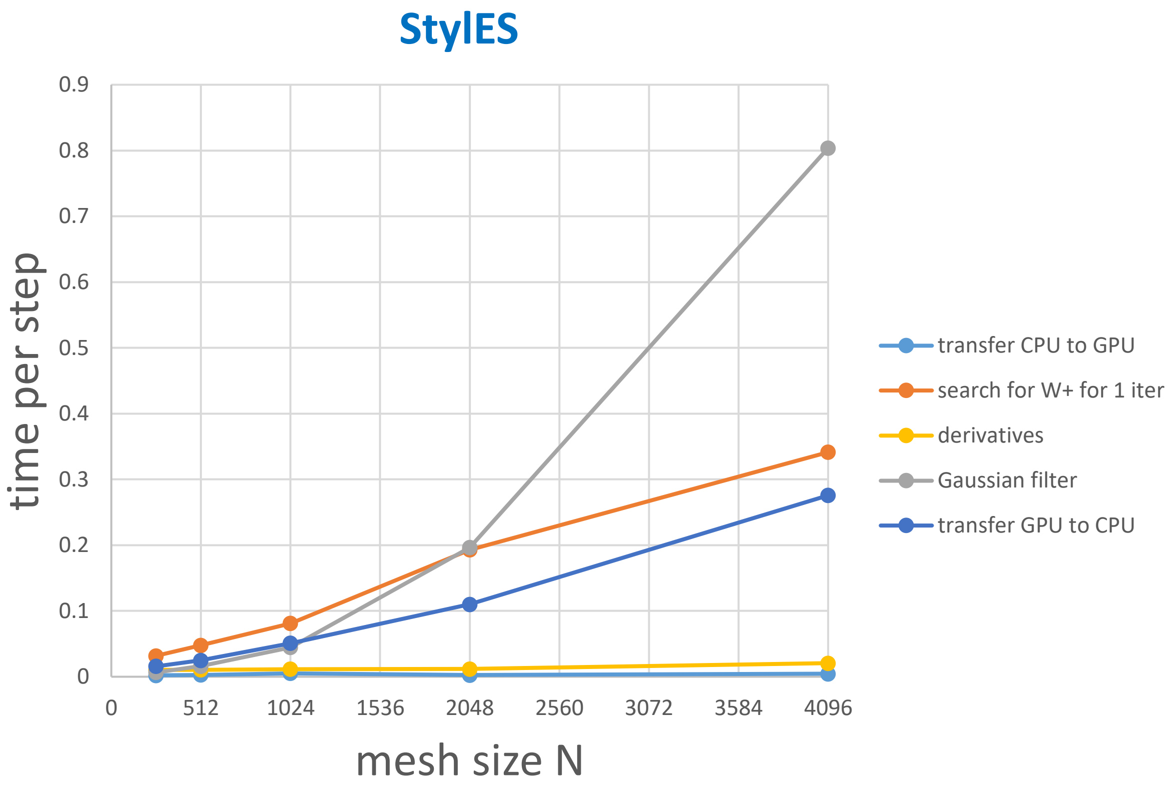
- faster LES time integration
- faster transfer CPU-GPU
- faster derivatives

- longer search for W^+
- longer filtering

<u>Note: the inference time is irrespective from the LES resolution chosen!!</u>

Performance (II)

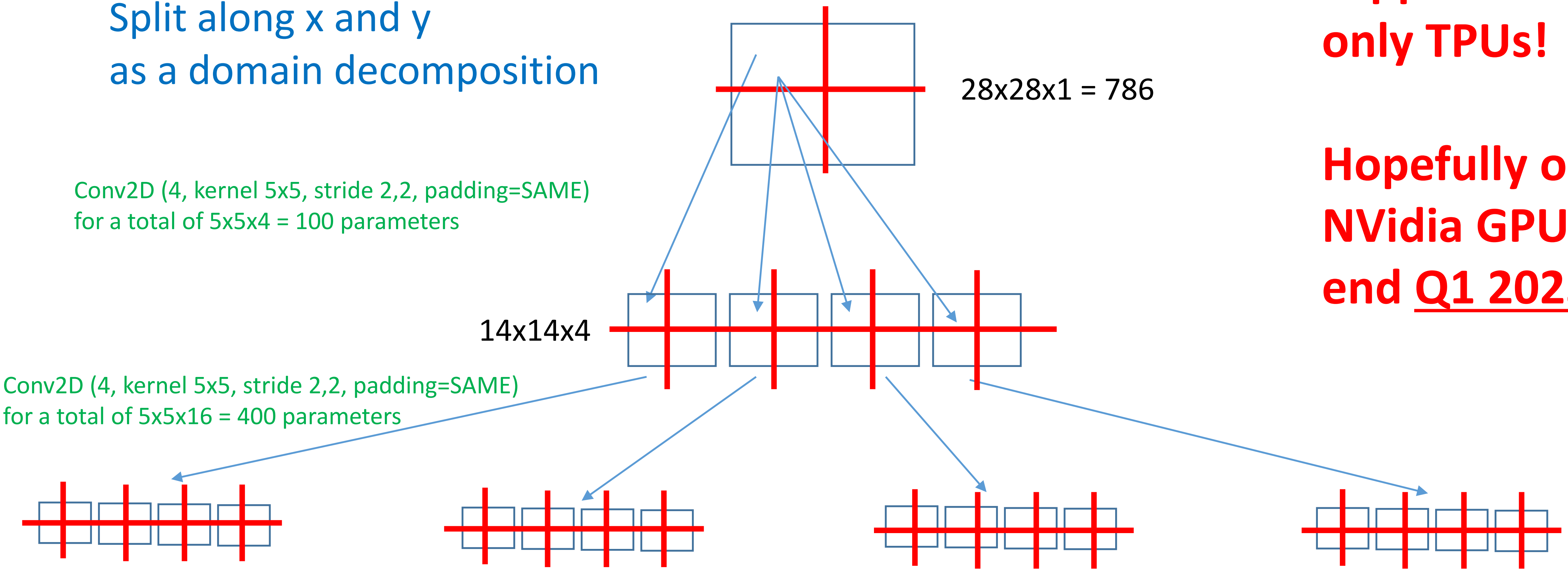




StylES is ~ 10% of the DNS!

Parallelization via DTensor (TensorFlow) of convolutional layers

Split along x and y
as a domain decomposition



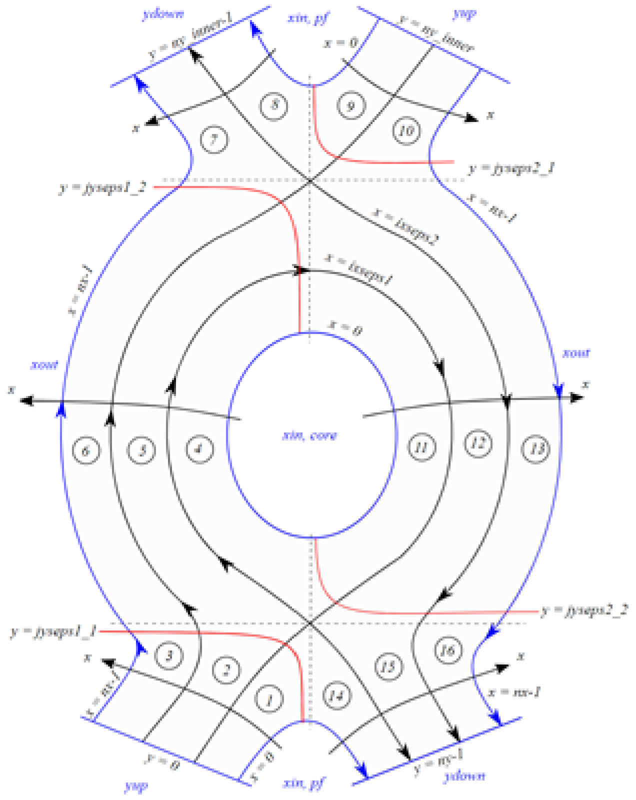
Currently supported only TPUs!

Hopefully on NVidia GPU by end Q1 2023!

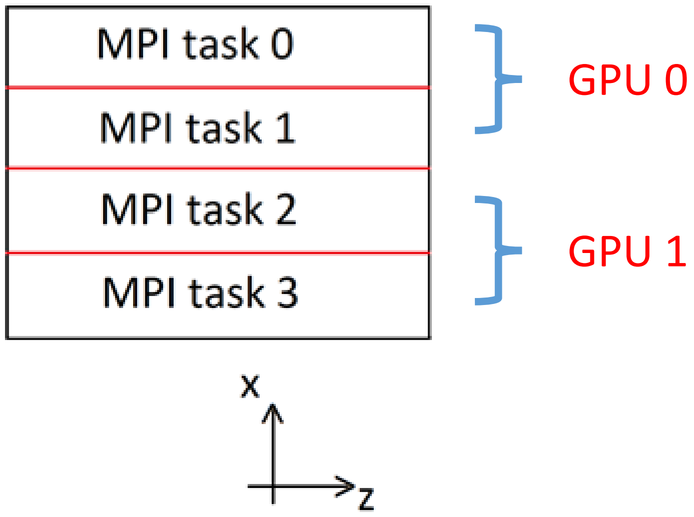
Science and Technology Facilities Council
Hartree Centre

Parallelization with BOUT++ and DTensor

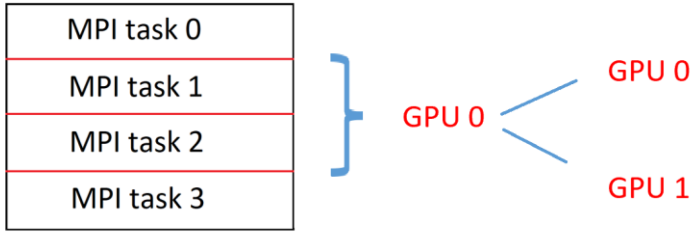
BOUT++ topology and parallelization
is quite complex…



…but, “Pencil decomposition” is also
supported in DTensor!



However, the split on Dtensor
currently happens from 1 GPU…



Science and Technology Facilities Council
Hartree Centre

Resume (I)

- We introduced a novel surrogate model based on latest Generative Adversarial Networks (GANs) for turbulent flow simulations
- This allows to avoid the train of a RNN for a time integration
- We do not use physic constrains yet, as these are inherited via the filter operator
- Good results obtained for HIT-2D, HW and mHW test cases
- Integration with BOUT++ is nearly completed, but some issues are currently encountered

Future work

- Complete integration StylES in BOUT++ and compare with LES models
- We need to optimize and parallelize the integration to multiGPU
- Extension to 3D (as a series of 2D planes along z…)