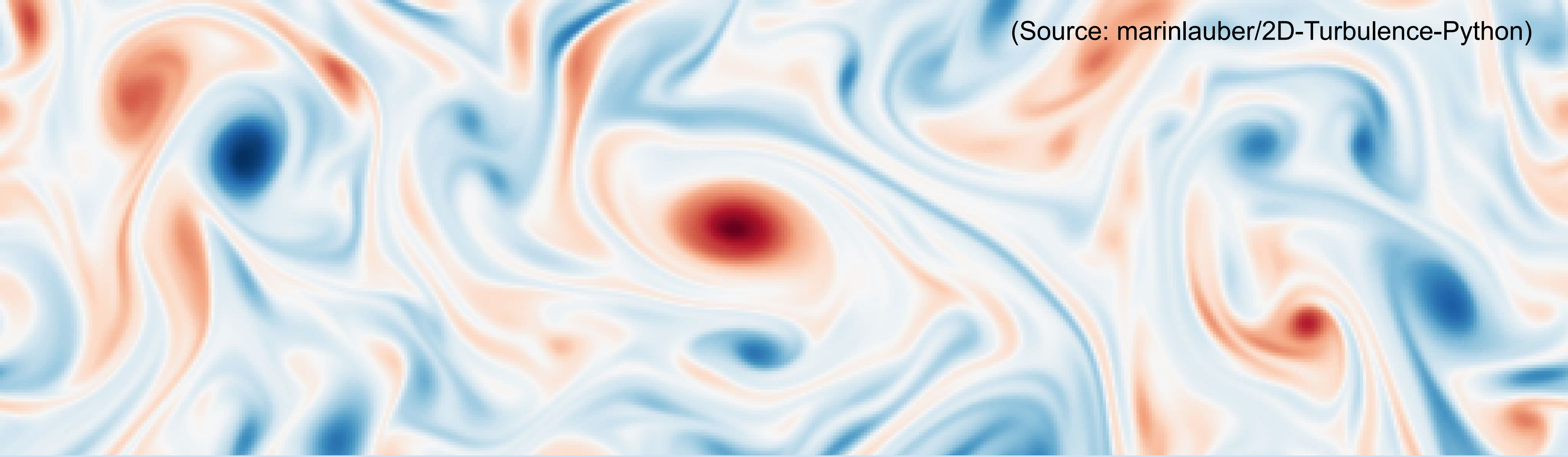


(Source: marinlauber/2D-Turbulence-Python)

Capturing Fine-Scale Turbulence in Planet-Forming Disks Using CNN-Augmented Fluid Solvers

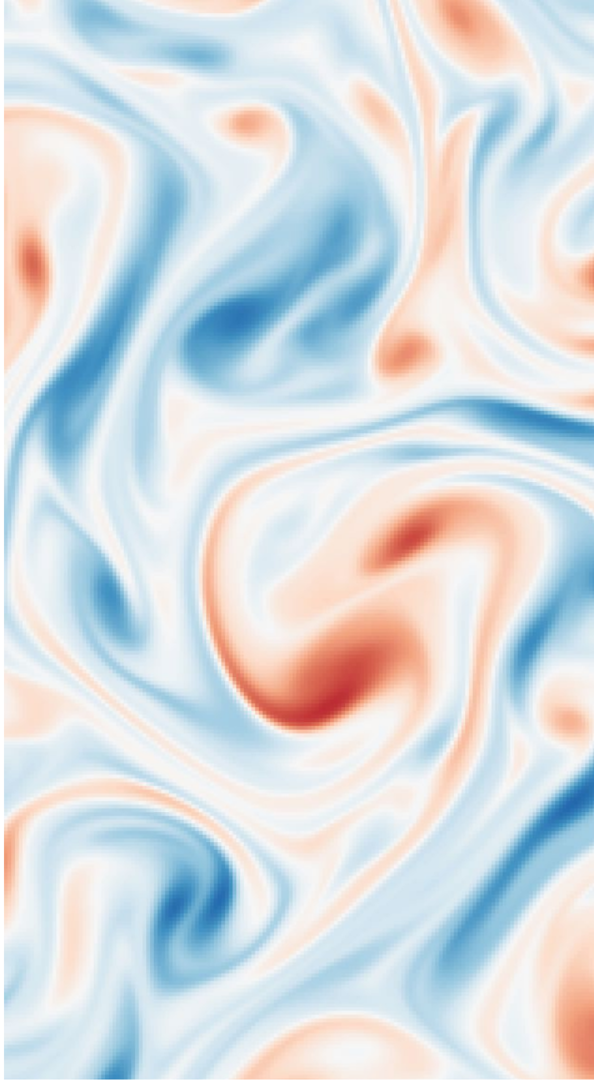
Amelie Lam, Héloïse Méheut, André Ferrari

Université Côte d’Azur, Observatoire de la Côte d’Azur, CNRS, Laboratoire Lagrange, Nice, France

ASNUM 15/12/2025

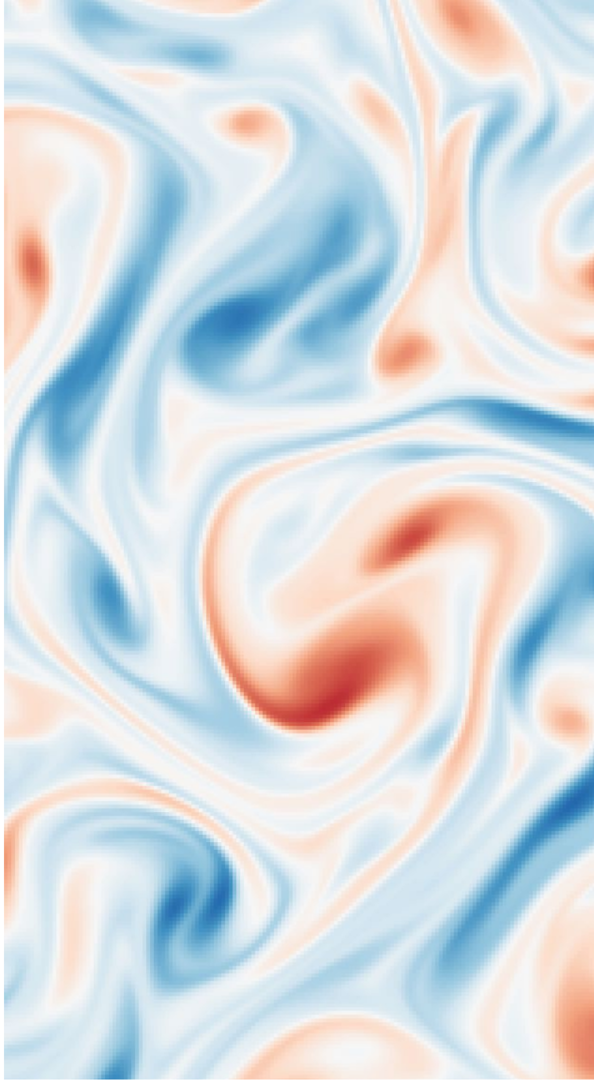
Outline

1. Problem Setup
2. Methodology
3. Metrics and Results
4. Conclusion



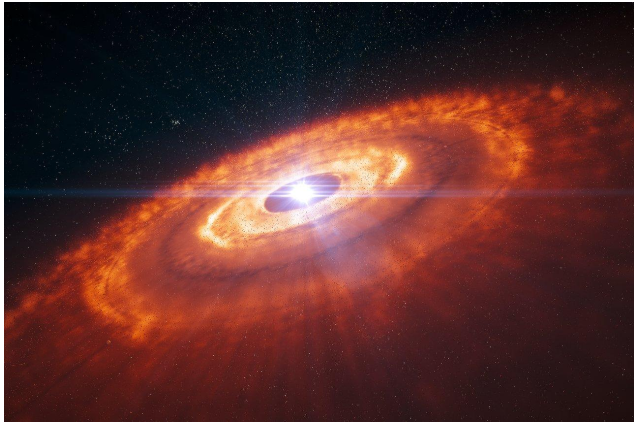
Outline

1. Problem Setup
2. Methodology
3. Metrics and Results
4. Conclusion

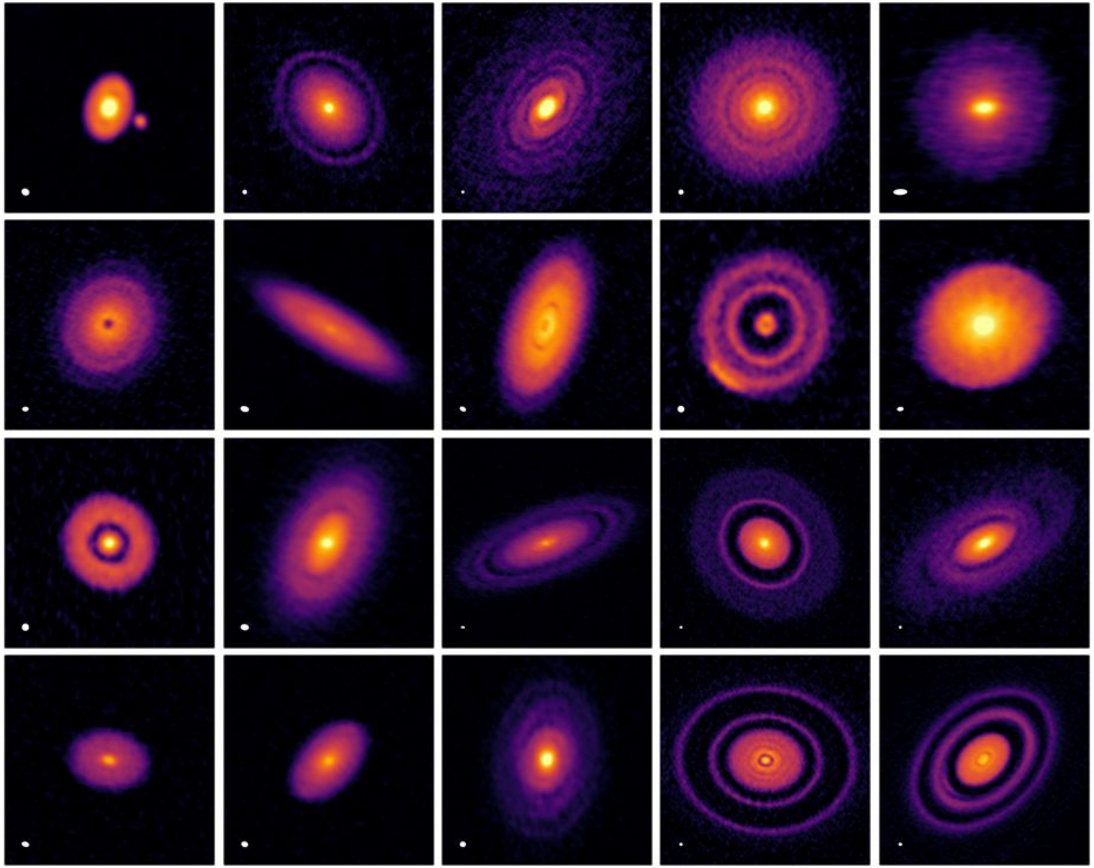


Planet-Forming Disk

- Keplerian disk around young star
- 99% gas, 1% dust



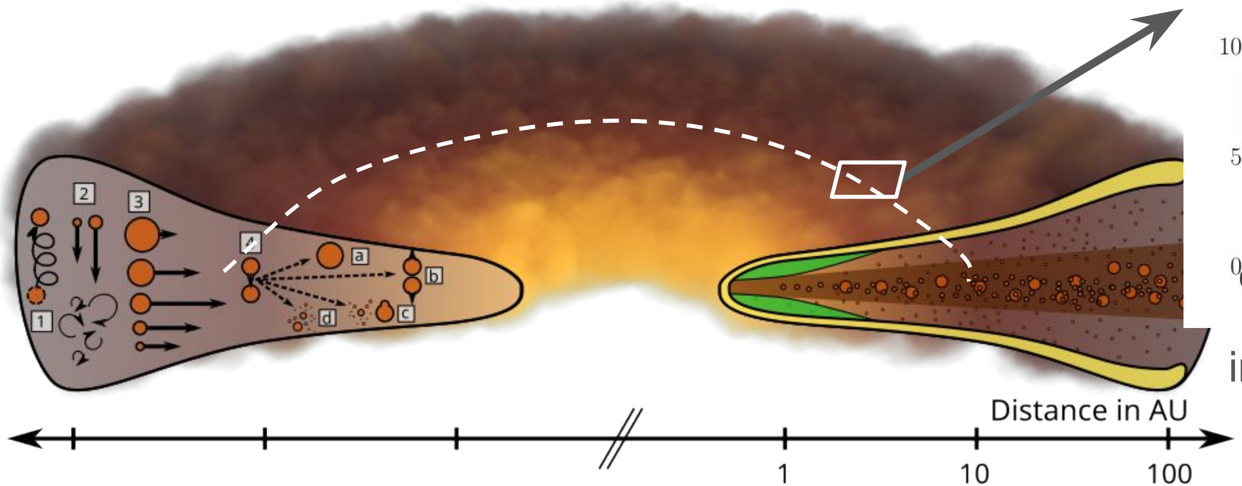
(Artist's Impression / ESO)



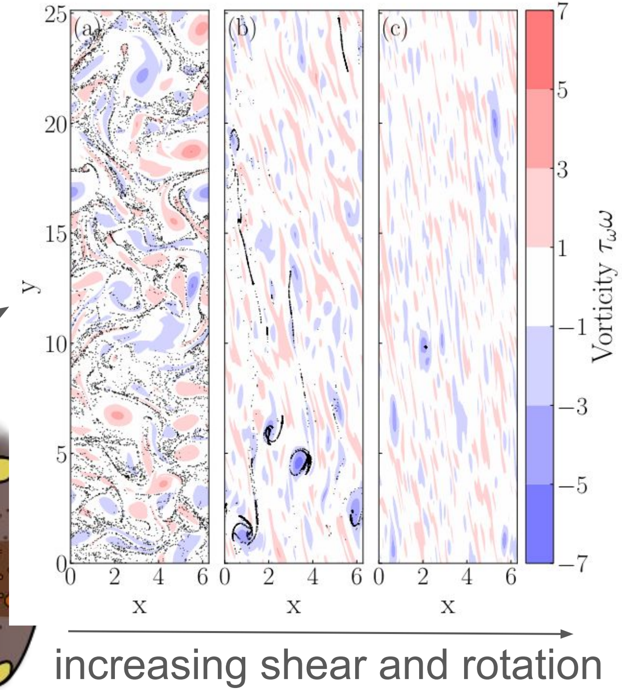
(Sean Andrews / DSHARP / ALMA)

Turbulence in Planet-Forming Disk is a Multiscale Problem

$$\boxed{\nu_T = \alpha c_s H}$$

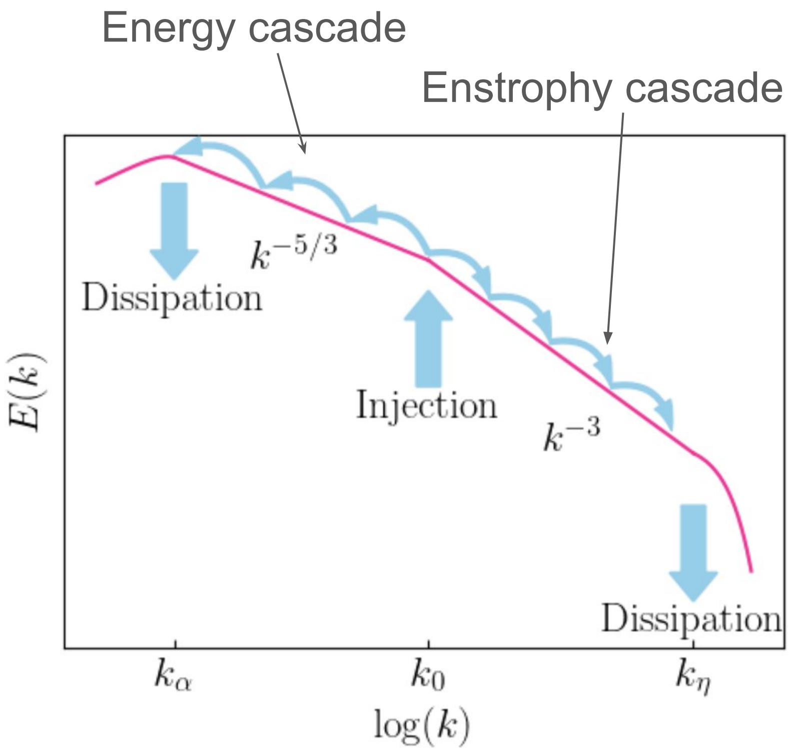


(Modified from Testi et al. 2014)



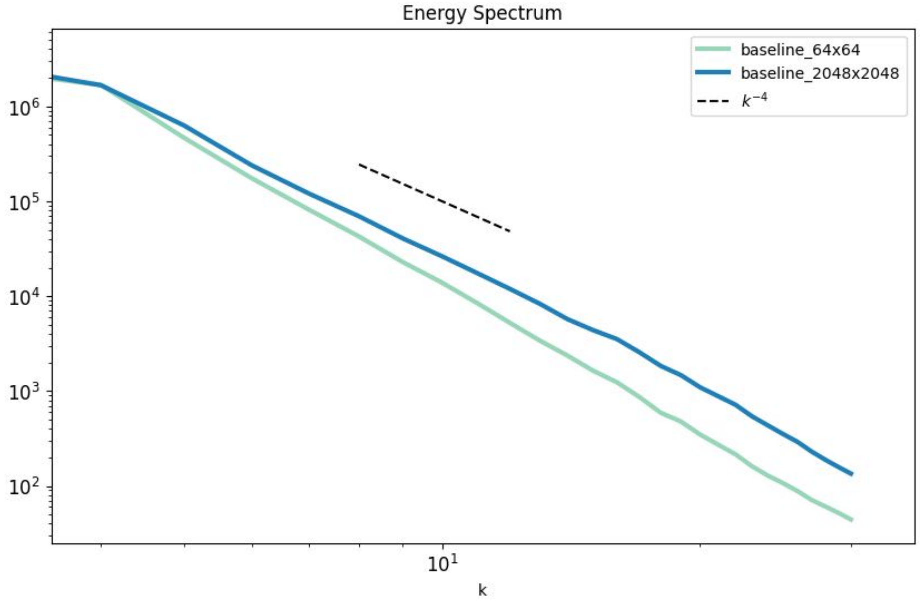
(Gerosa et al. 2023)

Turbulence Cascade



High Resolution —

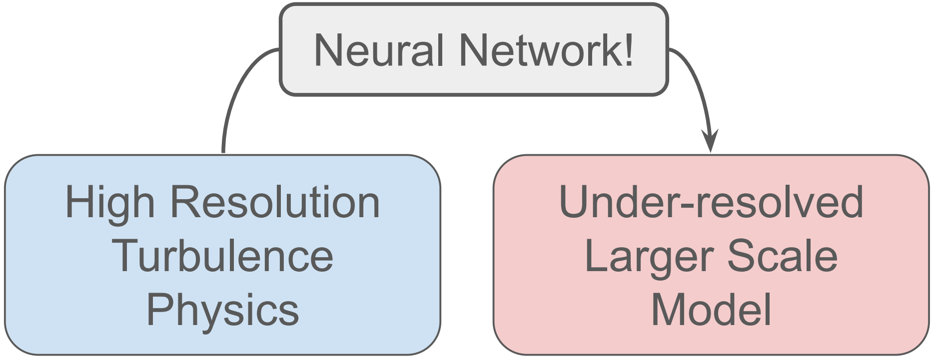
Low Resolution —



(Modified from Kochkov et al. 2021)

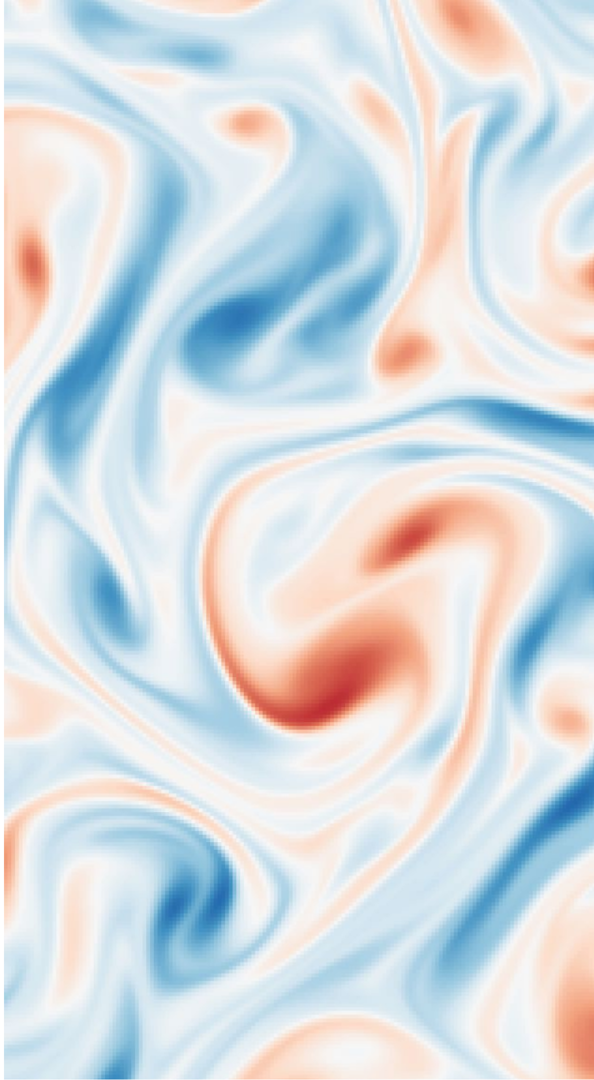
What we want to do?

1. Utilise suitable deep learning technique to capture small scale effect of turbulence inside a numerical solver
2. Produce fluid simulations that can resolve the effect of turbulence
3. Reduce computation cost



Outline

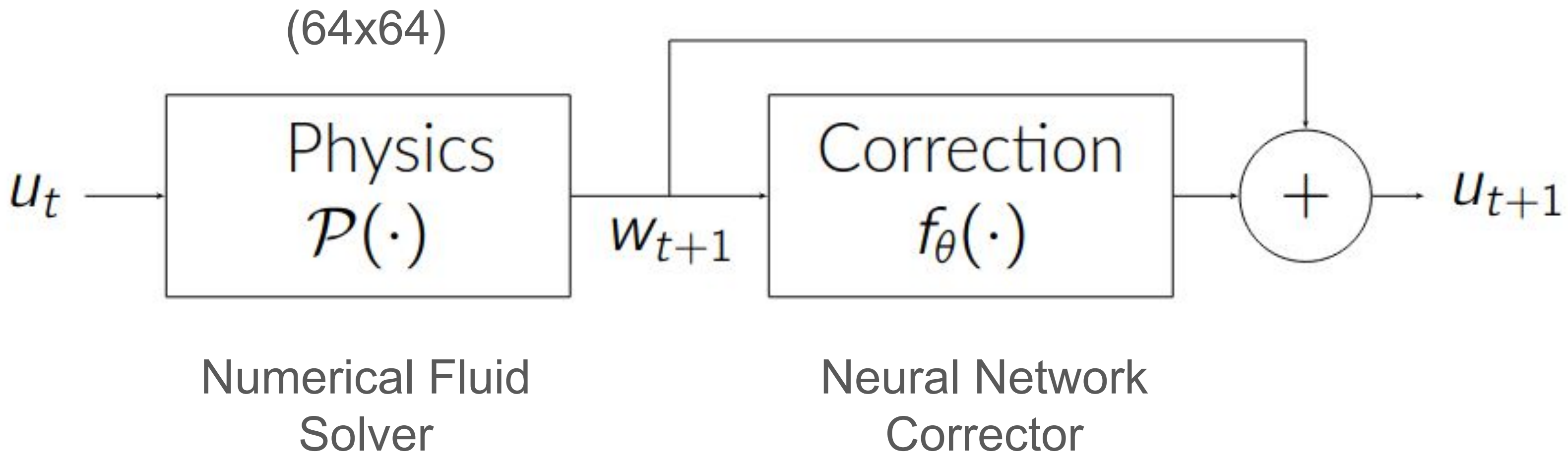
1. Problem Setup
2. Methodology
3. Metrics and Results
4. Conclusion



Our Proposed Solution

Inspired by Kochkov et al. 2021

For each timestep: $$u_{t+1} = \mathcal{P}(u_t) + f_\theta(\mathcal{P}(u_t))$$



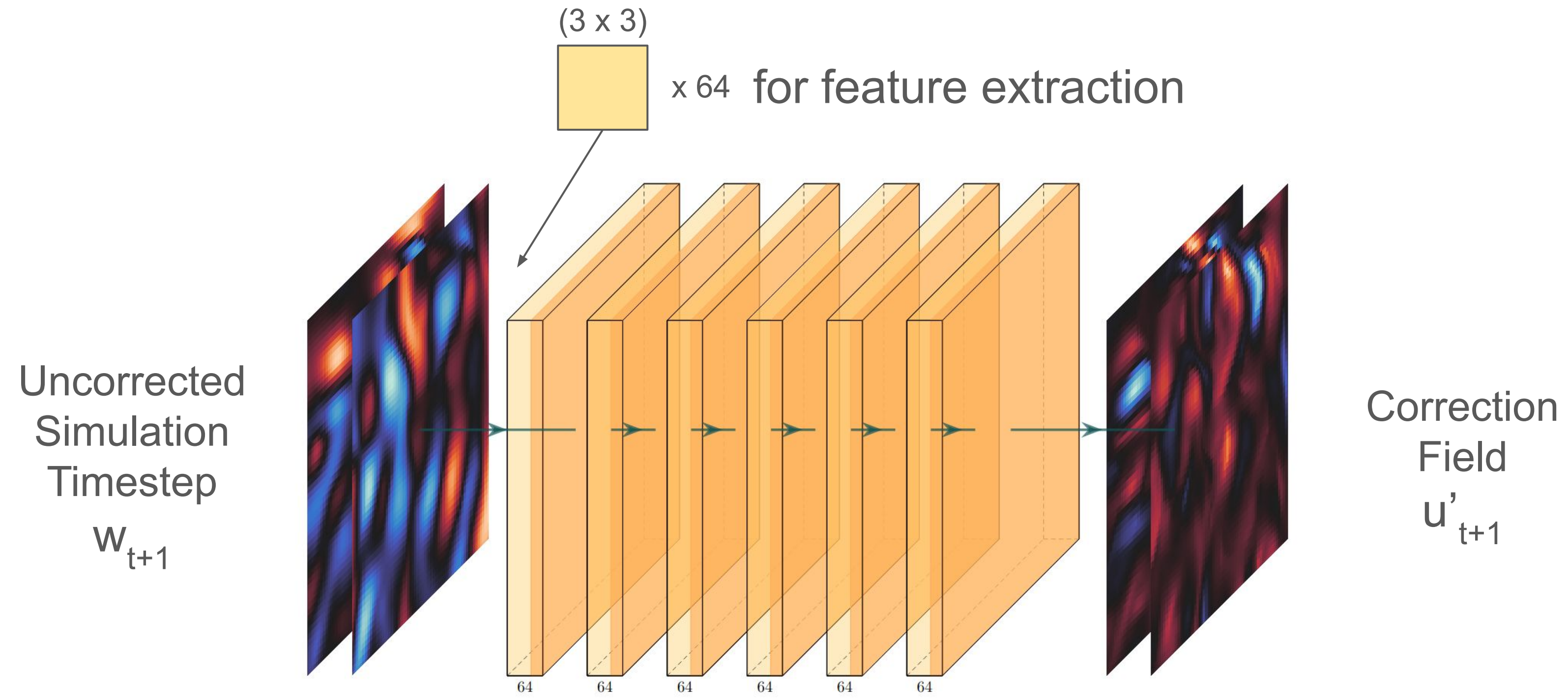
To model subgrid information and speed up simulation

What is important in a dynamic system?

Space! Time! Physics!

Space!

Convolutional Neural Network Corrector



Parameter count: 225K

How can we “teach” the neural network the correct dynamics?

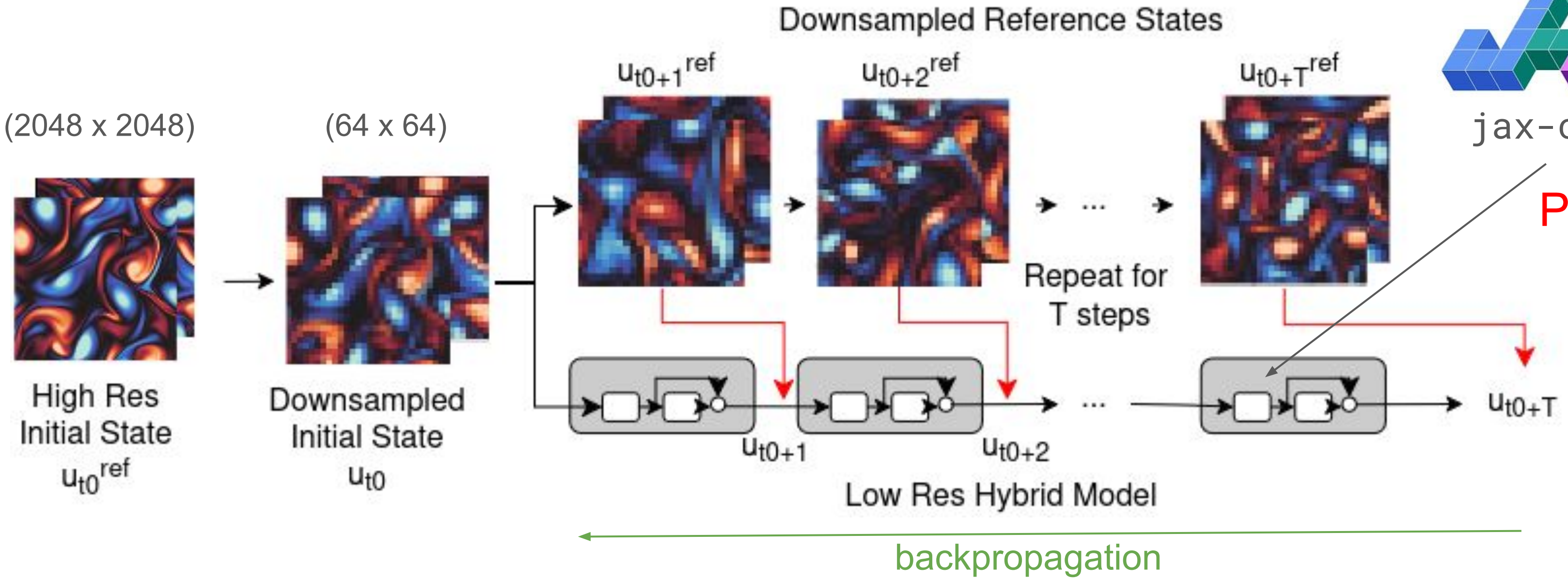
Time!

Optimizing over $$\mathcal{L}(\theta) = \sum_{\ell=1}^{M} \sum_{n=1}^{N-T} \sum_{i=n}^{n+T} (u_{i,\ell} - u_{i,\ell}^{\text{ref}})^2$$

“Solver-in-the-loop”

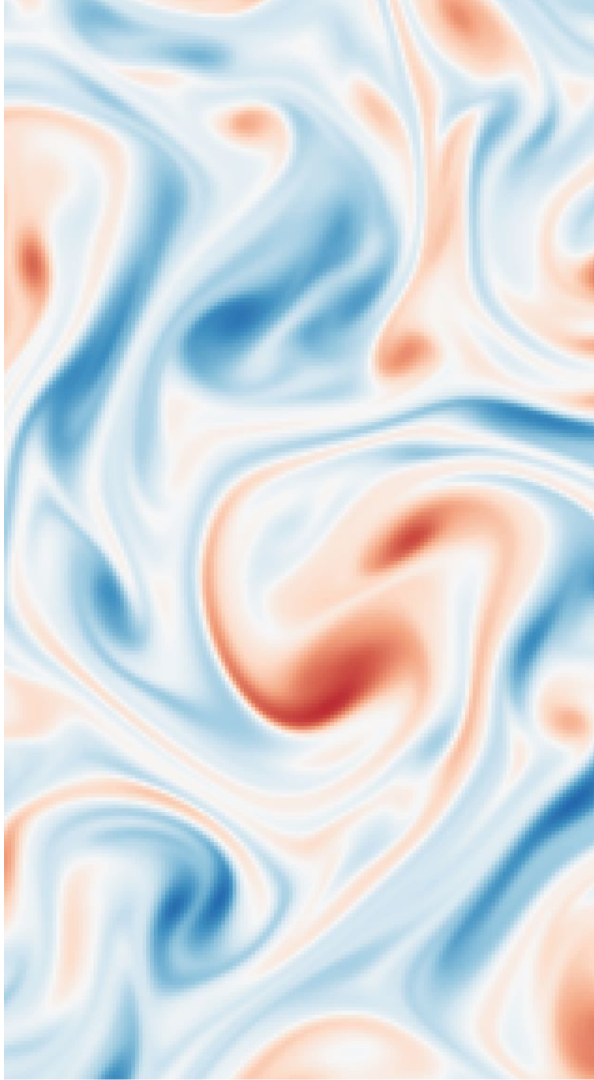
jax-cfd

Physics!



Outline

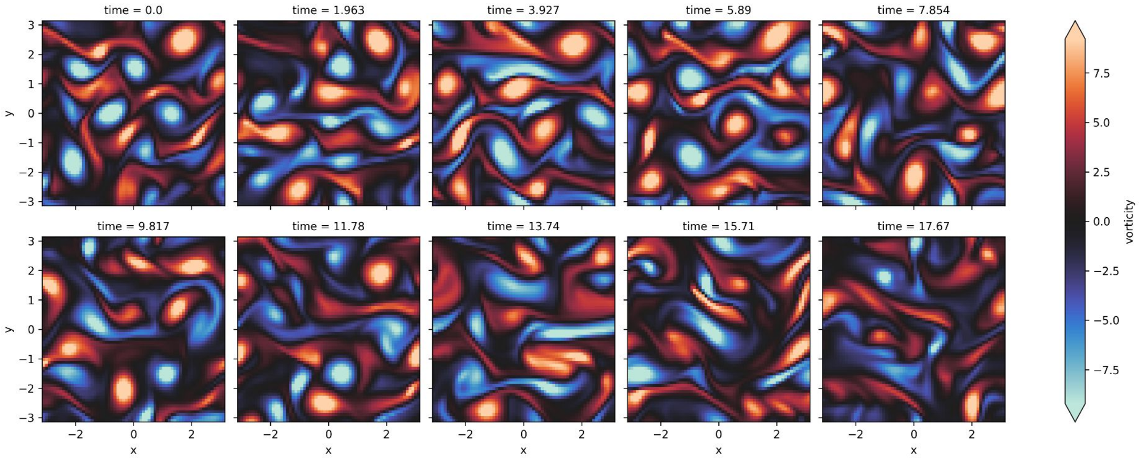
1. Problem Setup
2. Methodology
3. Metrics and Results
4. Conclusion



Test

Case Study: forced 2D incompressible homogeneous isotropic turbulence (HIT)

Dataset: 8 simulations of 500 timesteps initiated with different seeds (~7500 training samples)

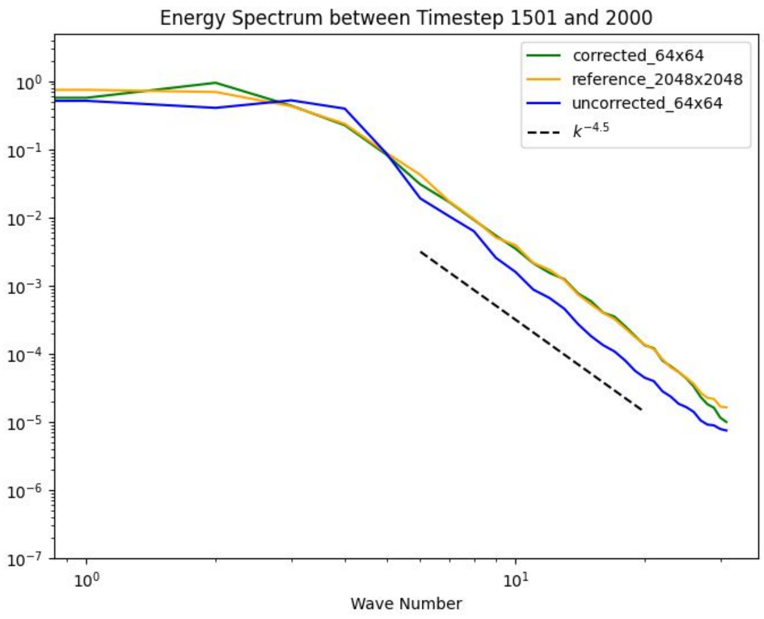


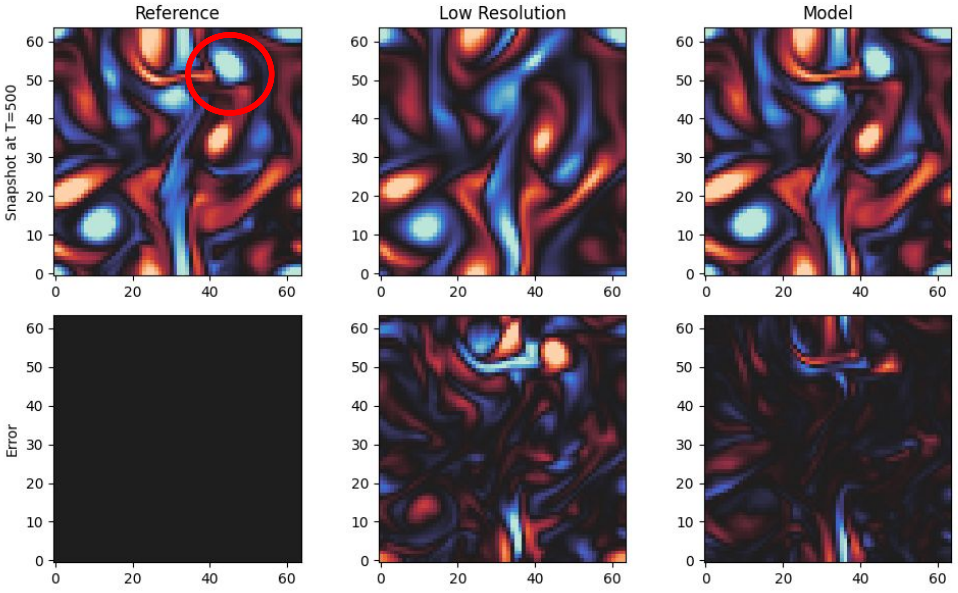
Test: simulations ran for 2000 steps (>> T)

Metrics: vorticity correlation, energy spectrum, vorticity field visualisation

Results

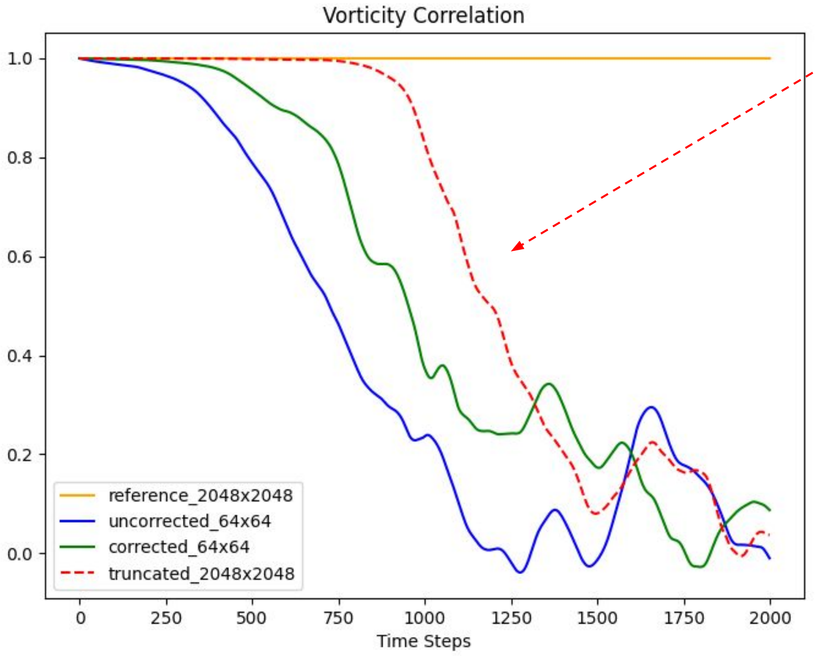
T=32, Epoch=400

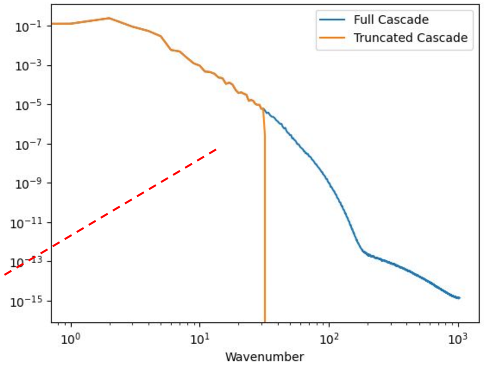


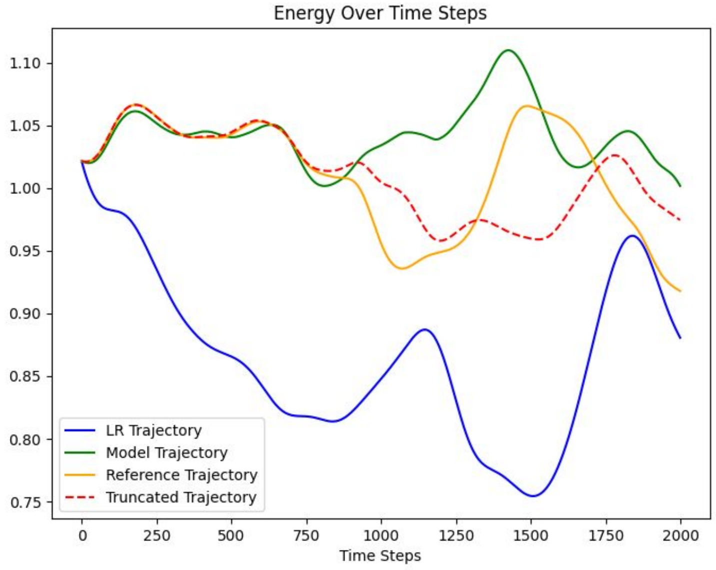


Timestep t=500

Results


Vorticity Correlation
1.0
0.8
0.6
0.4
0.2
0.0
0
250
500
750
1000
1250
1500
1750
2000
Time Steps
reference_2048x2048
uncorrected_64x64
corrected_64x64
truncated_2048x2048


Full Cascade
Truncated Cascade
Wavenumber


Energy Over Time Steps
1.10
1.05
1.00
0.95
0.90
0.85
0.80
0.75
0
250
500
750
1000
1250
1500
1750
2000
Time Steps
LR Trajectory
Model Trajectory
Reference Trajectory
Truncated Trajectory

Computation Time

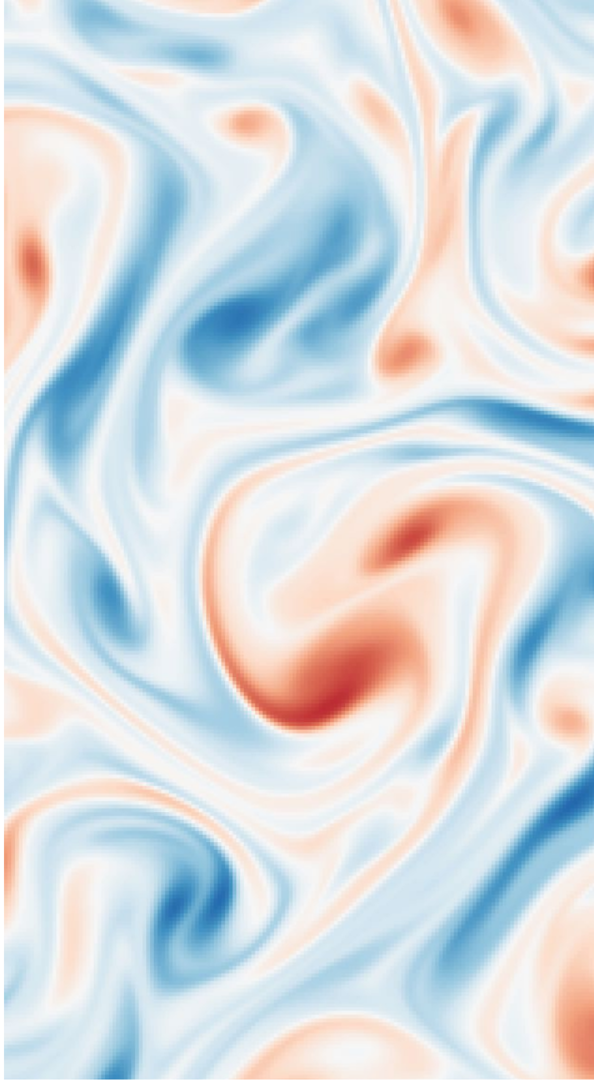
Model training Time: ~6 hours

Type of Simulation	Computation Time to Reach Simulation Time t = 70
High Resolution (2048 x 2048)	~ 7 minutes
Low Resolution (64 x 64)	~ 2 seconds
Low Resolution with Correction	~ 6 seconds

speedup
~x70

Outline

1. Problem Setup
2. Methodology
3. Metrics and Results
4. Conclusion



Conclusion

- Neural network is able to supply fluid solver with subgrid information
- Approach made possible with differentiable fluid solver
- Achieved a speed-up of ~x70 from high resolution simulations

Next steps:

- Modify fluid solver to enable shearing box simulations
- Improve neural network architecture
- Add particles into current framework
- Deploy model to existing codes


Thank you for listening!