# Embedding Hard Physical Constraints in Convolutional Neural Networks for 3D Turbulence


**Dr. Arvind T. Mohan**

**Postdoctoral Researcher**

Center for Nonlinear Studies

*Computational Physics & Methods Group*

*Los Alamos National Laboratory, New Mexico*




*Valles Caldera National Preserve*

# Acknowledgements


*Nick Lubbers*

*Information Sciences Group/LANL*

*Daniel Livescu*

*Computational Physics & Methods Group/LANL*

*Michael Chertkov*

*Dept. of Mathematics, University of Arizona*


UNCLASSIFIED

# Motivation

**Primary focus on the domain specialist end-users**. What do they want from a DL / statistical/ <insert your favorite> model?

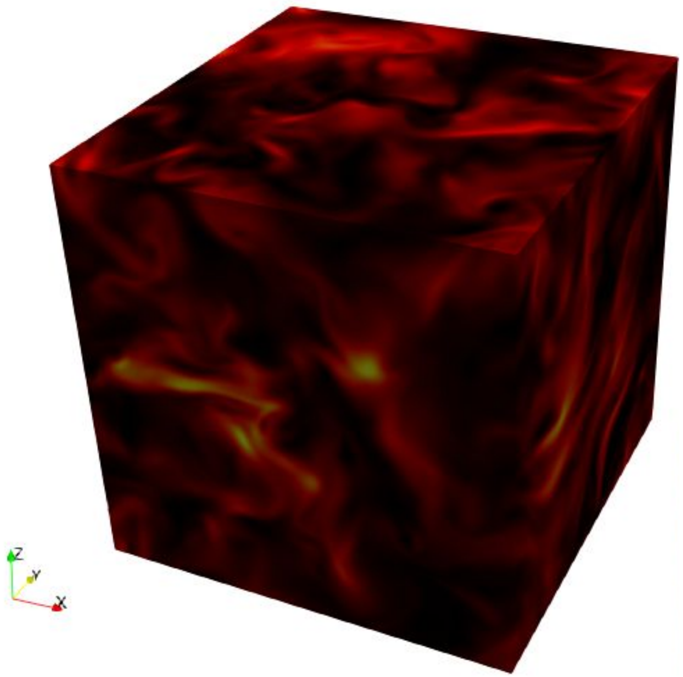
- Improved Accuracy
- Maximum interpretability / Intuition = consistent physics
- Robustness
- Developed on real world physics (very challenging)

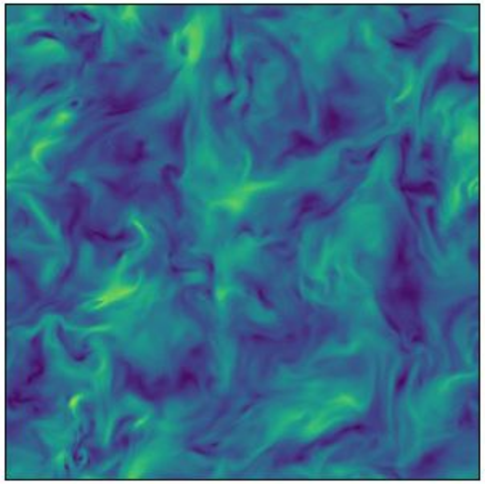
**Our philosophy:**

- Satisfy physics in DL model **by design** with inductive bias.
- **Add transparency** to black box DL models.
- Strive for better accuracy , BUT **trade-off** with interpretability + robustness.
- Need simple dataset to develop algorithm, but need to retain realism:
  **Use 3D, fully developed, turbulence**

# Test Case: Homogenous Isotropic Turbulence (HIT)

- DNS dataset of HIT in a cube – stationary in time. Periodic boundary conditions
- Goal: Learn spatio-temporal 3D dynamics from few snapshots Domain Size: $128^3$
- Training Data: 0 – 1 eddy time. Test Data: > 1.5 eddy times.



Incompressible flows are "divergence-free", Can we…

1) **Guarantee divergence-free inductive-bias in the CNN regardless of training hyper-parameters?**

2) **Guarantee boundary conditions always enforced?**

Instead of loss functions, we directly **embed mass conservation law** into network architecture

$$V = \nabla \times A + \nabla\psi$$

$$\nabla \cdot V = 0$$

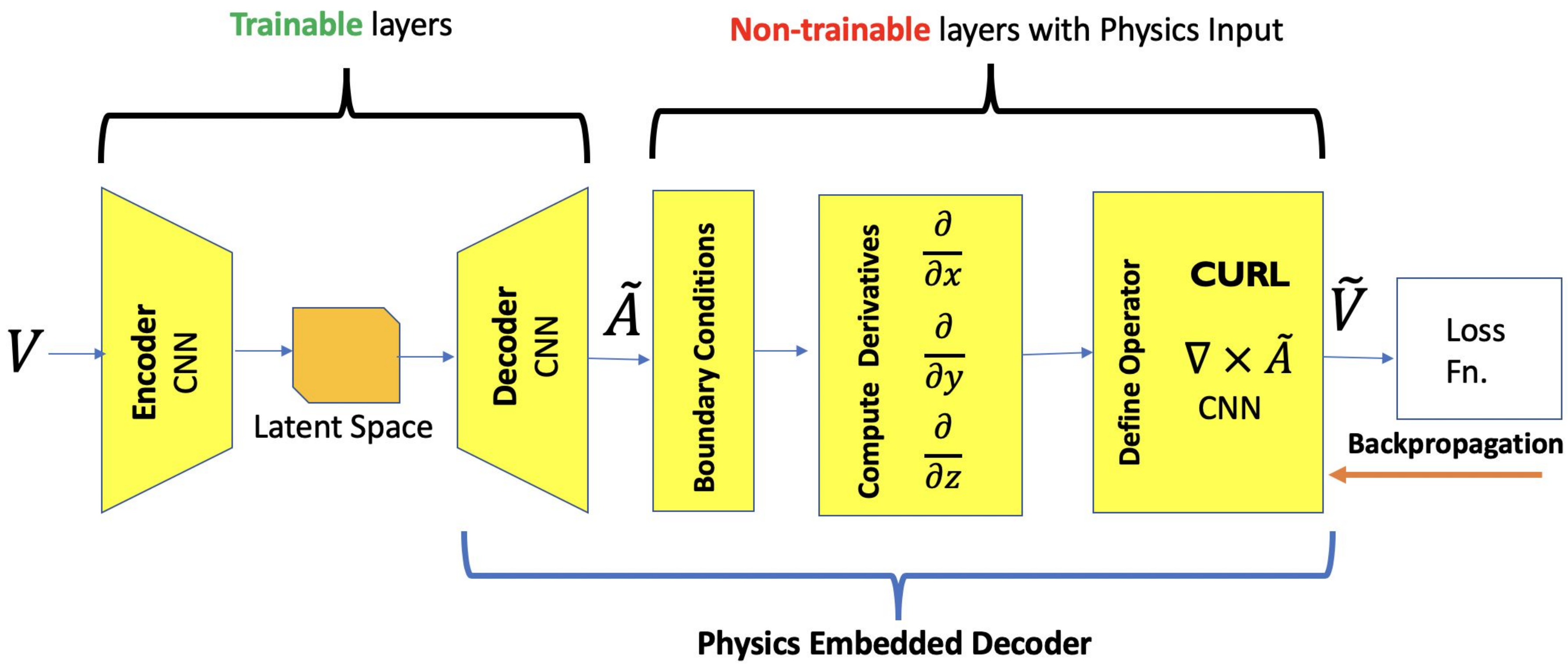
A is **potential vector field**
U is **velocity field**

$\nabla\psi = 0$ for periodic BCs in HIT simulation

If we can **learn** $V$ as $V = \nabla \times A$, we **automatically satisfy** Divergence free condition $\nabla \cdot V = 0$

**Strategy:** Embed the **boundary conditions** and **CURL operator** $\nabla \times$ in the autoencoder network.

# *Physics-Embedded* Convolutional Autoencoder for 3D flow (PhyCAE)




$\tilde{A}$ is reconstructed vector potential field – let network "choose" appropriate gauge!

# Injecting Differential Operators into CNN

Need a method that is time-tested, interpretable,
And already used in production…….. *Numerical Methods*

In PDE Finite Difference (FD)/ Volume discretized (FV) solvers **also perform** convolution of a kernel $f$ with the spatial domain $g$

$$h = f * g$$

$f$ is **differential operator** kernel → from Taylor Series approximations of expressions (approximated to numerical order)

$g$ is the solution residing on a mesh

$$\frac{\partial \phi}{\partial x} = \frac{\phi(x+\delta x)-\phi(x-\delta x)}{2\delta x} + O(\delta x)^2$$

$$\frac{\partial \phi}{\partial y} = \frac{\phi(y+\delta y)-\phi(y-\delta y)}{2\delta y} + O(\delta y)^2$$

$$\frac{\partial \phi}{\partial z} = \frac{\phi(z+\delta z)-\phi(z-\delta z)}{2\delta z} + O(\delta z)^2$$

Kernel form □

$$h_1 = \frac{\partial}{\partial x} * \phi$$

$$h_2 = \frac{\partial}{\partial y} * \phi$$

$$h_3 = \frac{\partial}{\partial z} * \phi$$

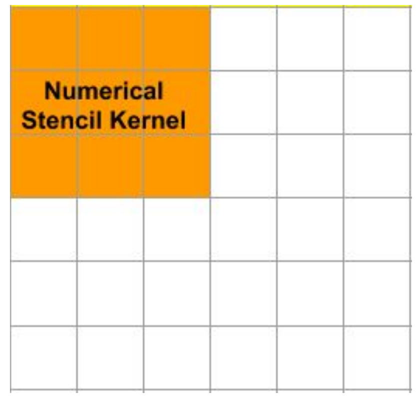
FV stencil for 2nd order Central differencing

# FD/FV Stencils □□ Convolutional Network Kernels

$$\frac{\partial \phi}{\partial x} = \begin{bmatrix} 0 & 0 & 0 \\ -\frac{1}{2\delta x} & 0 & \frac{1}{2\delta x} \\ 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial \phi}{\partial y} = \begin{bmatrix} 0 & -\frac{1}{2\delta y} & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{2\delta y} & 0 \end{bmatrix}$$

$$\frac{\partial \phi}{\partial z} = \begin{bmatrix} 0 & 0 & 0 \\ -\frac{1}{2\delta z} & 0 & \frac{1}{2\delta z} \\ 0 & 0 & 0 \end{bmatrix}$$




$Stride = 1$ in CNN
with differential kernels
computes $h_1$, $h_2$, $h_3$
To numerical accuracy

Long et. Al. - PDE-Net (2018)
Dong et. Al. (2017)

- Spatial derivatives → Convolutional kernel of constant weights in FDM/FVM s
- Backpropagate thru operators defined by derivatives *e.g.* $\nabla \times \tilde{A}$, $\nabla \cdot$
- Easier book-keeping, can utilize existing parallel/distributed training infrastructure for CNNs in Pytorch/Tensorflow

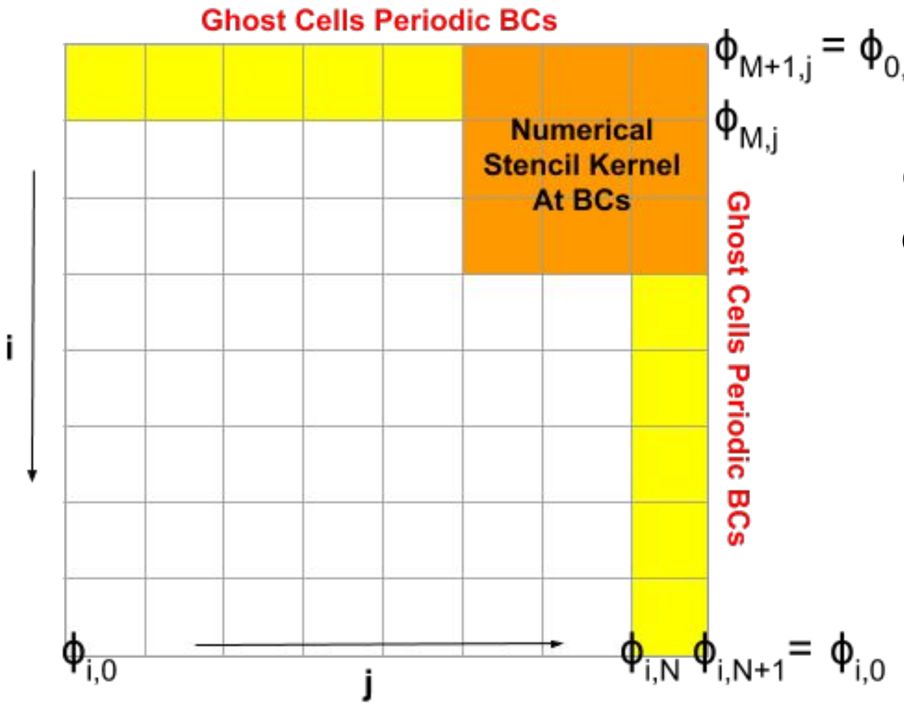
BUT still need boundary conditions to be strictly enforced…..

# Consistent Boundary Conditions in CNN

Like PDE solvers, ensure BCs are always present during training, and not minimize as a constraint

Solution: **Ghost Cell approach from CFD.** Established approach in community!

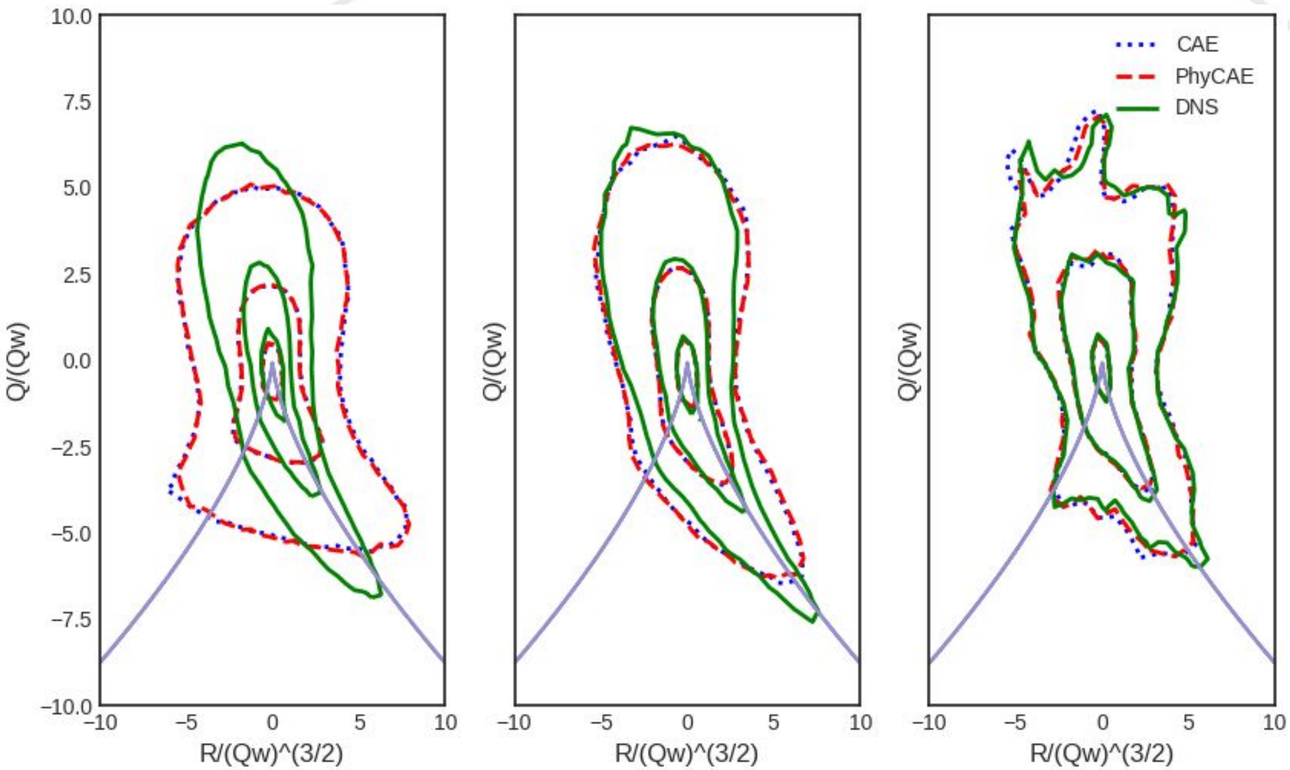
Instead of zero/reflection padding □ Build custom padding to enforce periodicity with Ghost cells




Can increase/decrease ghost cells for desired order of accuracy with FV numerical stencil

$$\phi_{i,N+1,k} = \phi_{i,0,k}$$
$$\phi_{i,N+2,k} = \phi_{i,1,k}$$
$$\phi_{i,N+3,k} = \phi_{i,2,k}$$

# RESULTS: Q-R plane morphology of Small, Inertial and Large Scales – **Stringent test of 3D turbulence**



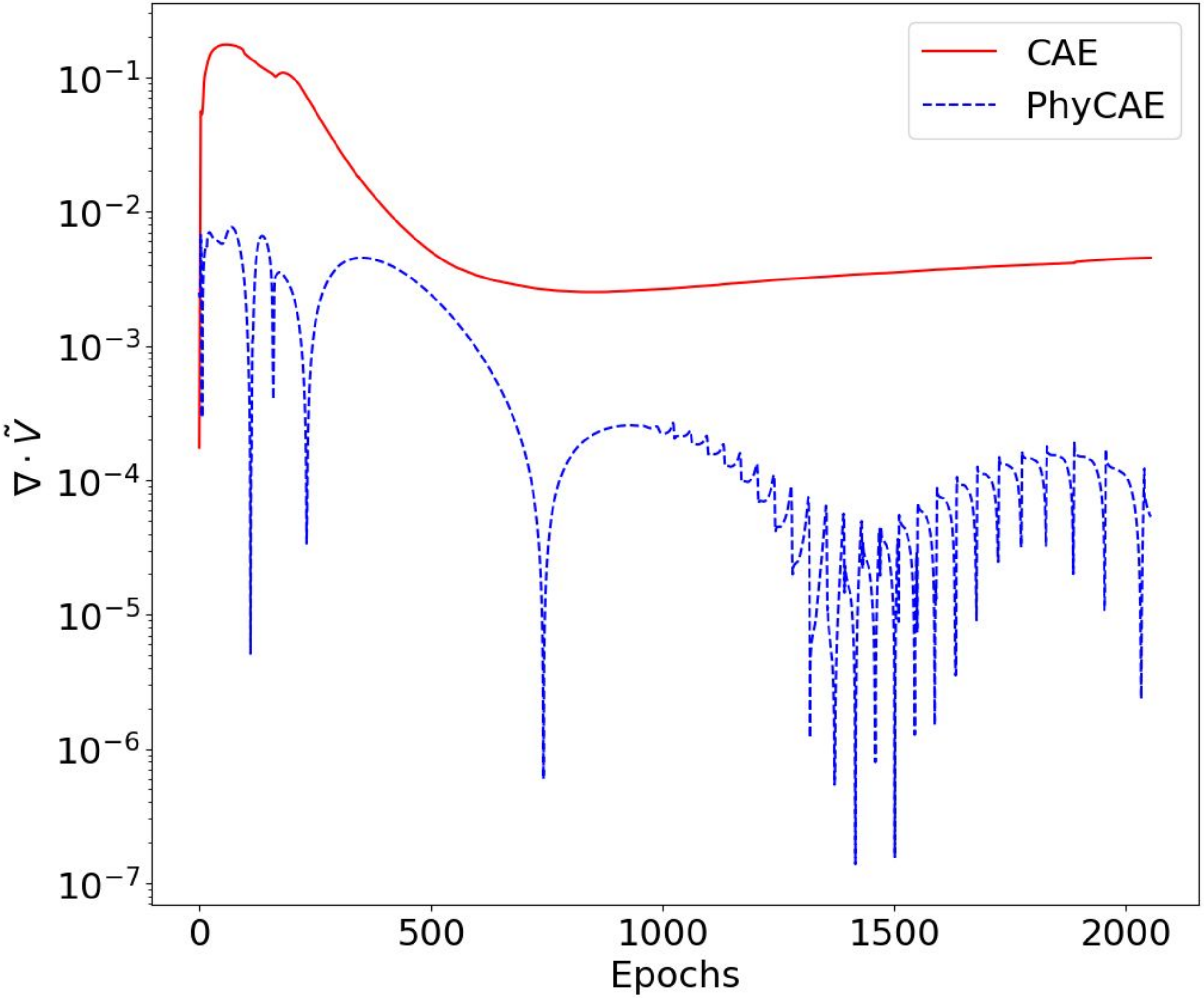

Small Inertial Large

Coarse-graining excellent accuracy for large scales : Small scales are largely neglected.

**Large scales critical for several applications**

*Compression ratio size(original)/size(latent space)* ***~ 300x***

# Learning: Unconstrained Network vs Physics Embedded Network




**Final test Divergence**

(Float32 computation)

**CAE**

$$\nabla \cdot \tilde{V} \approx 10^{-2}$$

**PhyCAE**

$$\nabla \cdot \tilde{V} < 10^{-4} - 10^{-5}$$

UNCLASSIFIED

# Summary

- ✔ Architecture integrates CFD/**numerical methods** with CNNs for embedding mass conservation.
- ✔ General framework to embed boundary constraints and compute various operators as a CNN, with desired Finite Volume/Finite Difference schemes
- ✔ **No increase** in trainable parameters compared to the generic, unconstrained network.
- ✔ Useful when we don't have the full governing equations, but only know constraints.
- ✔ Architecture with **strong inductive bias for** incompressible flow: **More Interpretable**

General strategy to learn 3D fields with constraint of form

$$L(V) = G$$

Where $L$ is a differential operator

A Mohan, N. Lubbers, M Chertkov, D. Livescu arXiv: 2002.00021

# Thank you!

arvindm@lanl.gov

@ArvindMohan15

*Rio Grande River Los Alamos, NM*