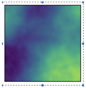
Learning Operators

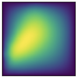
Siddhartha Mishra

Computational and Applied Mathematics Laboratory (CamLab)
Seminar for Applied Mathematics (SAM), D-MATH (and),
ETH AI Center,
ETH Zürich, Switzerland.

Why Operators ?

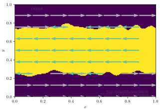
- Operator: $\mathcal{G} : \mathcal{X} \mapsto \mathcal{Y}$, $\dim\,(\mathcal{X}, \mathcal{Y}) = \infty$.
- Example: Hamiltonian Operators of Quantum Physics
- Example: Solution Operators of PDEs.
- Darcy PDE: $-\mathrm{div}(a\nabla u) = f$, $\mathcal{G} : a \mapsto \mathcal{G}a = u$.

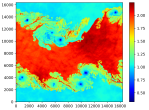




a $\quad$ u

- Compressible Euler equations: $\mathcal{G} : u_0 \mapsto \mathcal{G}u_0 = u(t)$.

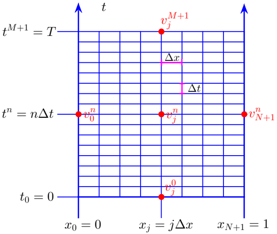




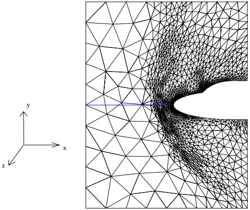
Initial Condition $\quad$ Solution at time T

- Solving (approximating) a PDE is approximating an operator

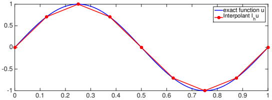
Traditional Numerical Methods



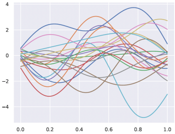
Finite Difference



Finite Volume



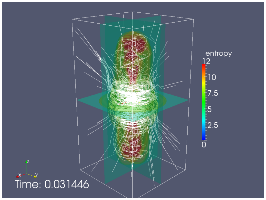
Finite Element



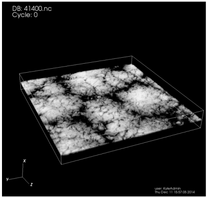
Spectral Method

- Different flavors of Runge-Kutta for Time Integration

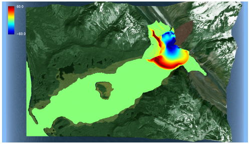
Numerical Methods are very Successful



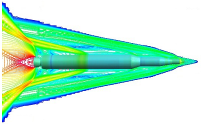
Supernovas



Clouds

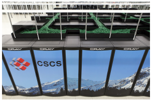


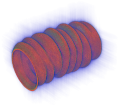
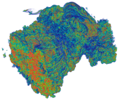
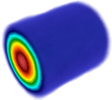
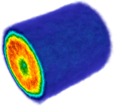
Tsunamis



Rockets

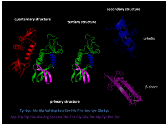
Issues with Traditional Numerical Methods

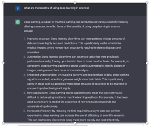


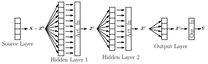
- Computational Cost !!: PDE solvers can be very expensive.
- Many-Query Problems: UQ, Design, Inverse Problems.
- Ensemble simulation of Navier-Stokes at 1024^3 costs ≈ 1
  - With Azeban on Piz Daint.
  - Single Sample: 94 node hours.
  - Ensemble simulation: 96256 node hours
  - Cost: Approx $560K$ CHF.
- Inefficient Data usage: Data discarded after simulation.
- AIM: Learn Operators from Data + Physics

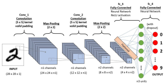
The age of Machine Intelligence

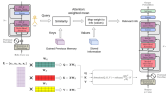




- 3 Pillars of Success !!
- Exponentially more Compute aka GPUs :-)
- Huge Data
- Deep Neural Networks: $\mathbb{R}^N \mapsto \mathbb{R}^M$

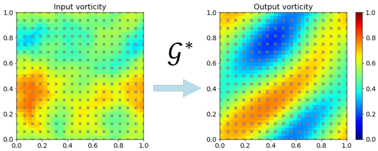




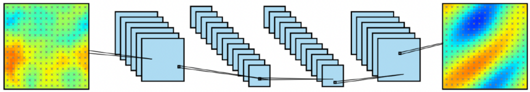


- Operator Learning: Inputs+Outputs are Functions !!!

An Obvious Solution

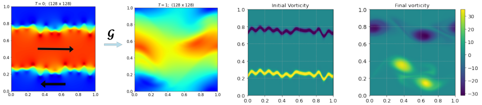


- Uniform Sampling $\mapsto$ CNN $\mapsto$ Interpolation



Does this work ?

- Shear flow with Navier-Stokes with $Re >> 1$

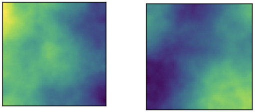


- CNN Results:

- Consistent with Zhu,Zabaras, Li et. al, Kovachki et. al, +++
- Desiderata for Operator Learning:
  - Input + Output are functions.
  - Some form of Resolution Invariance
- Learn underlying Operator, not just a discrete Representation

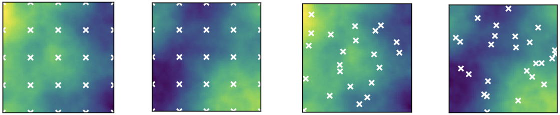
Why is this a challenge ?

- In principle, Operator maps functions to functions.



Input Output

- In practice, both inputs and outputs are Discrete

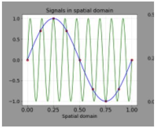


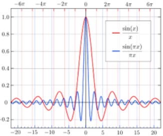
Input Output Input Output

- Multiple Discrete Representations !!
- Only discrete operations on Digital Computers.
- A proper notion of Continuous-Discrete Equivalence (CDE)

A Signal Processing Detour: CDE for Functions

- Can $f \in L^2$ be uniquely recovered from point values?



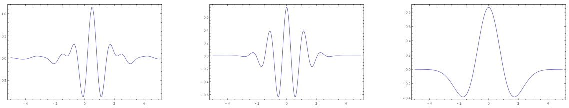


- Whittaker-Kotelnikov-Shannon Sampling Theorem:
- Yes, if $f \in \mathcal{B}_\Omega$ and grid size $\Delta \leq 1/2\Omega$
- $\mathcal{B}_\Omega$ is Bandlimited functions i.e., $\mathrm{supp}\ \hat{f} \subset [-\Omega, \Omega]$
- Unique representation in terms of basis of Sinc functions:

$$f = \sum_{n \in \mathbb{Z}} f(n\Delta)\varphi_n \quad \Leftrightarrow \quad \mathcal{P}_{\mathcal{B}_\Omega}(f) = f$$

- Lack of CDE $\Rightarrow$ Aliasing error: $\varepsilon(f) = f - \mathcal{P}_{\mathcal{B}_\Omega}(f)$

A Harmonic Analysis Detour: Frame Theory



- φ_i is a Frame in Hilbert space $\mathcal{H}$ if

$$A\|f\|^2 \leq \sum_i |\langle f, \varphi_i \rangle|^2 \leq B\|f\|^2, \quad \forall f \in \mathcal{H}.$$

- Synthesis: $T : \{c_i\} \mapsto \sum_i c_i \varphi_i$, Analysis: $T^* : f \mapsto \{\langle f, \varphi_i \rangle\}$
- Frame Operator: $S = TT^*$, $T^\dagger : f \mapsto \langle f, S^{-1}\varphi_i \rangle$
- Frame Reconstruction formula: $f = \sum_i \langle f, \varphi_i \rangle S^{-1} \varphi_i$
- CDE for subspace $\mathcal{V}$ of $\mathcal{H}$ in terms of Frame sequence v_i.
- Lack of CDE $\Rightarrow$ Aliasing error: $\varepsilon(f) = f - \mathcal{P}_{\mathcal{V}}(f)$

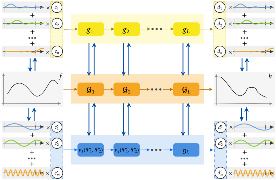
CDE for Operators: Bartolucci et al, Neurips 2023

$$\begin{tikzcd}
\mathcal{H} \arrow[r, "U"] \arrow[d, "T_\Psi^\dagger"] & \mathcal{K}, \\
\ell^2(I) \arrow[r, "u"] & \ell^2(K) \arrow[u, "T_\Phi"]
\end{tikzcd}$$

- Operator $U : \mathcal{H} \to \mathcal{K}$, with frame sequences: (Ψ, Φ)
- Aliasing Error operator: $\varepsilon(U, u, \Psi, \Phi) := U - T_\Phi \circ u \circ T_\Psi^\dagger$.
- $(U, u(\Psi, \Phi))$ is Representation Equivalent:
- $\varepsilon \equiv 0$, for all $\mathrm{Dom}\ U \subset \mathcal{M}_\Psi$ $\mathrm{Ran}\ U \subset \mathcal{M}_\Phi$
- Uniqueness of Discrete Representations

$$\begin{tikzcd}
 & \ell^2(I) \arrow[r, "\mathfrak{u}(\Psi,\Phi)"] & \ell^2(K) \arrow[dr, "T_\Phi"] & \\
\mathcal{H} \arrow[ur, "T_\Psi^\dagger"] \arrow[rrr, "U"] \arrow[dr, "T_{\Psi'}^\dagger"] & & & \mathcal{K} \\
 & \ell^2(I') \arrow[r, "\mathfrak{u}(\Psi',\Phi')"] & \ell^2(K') \arrow[ur, "T_{\Phi'}"] &
\end{tikzcd}$$

Representation Equivalent Neural Operators (ReNO)



- Layerwise Representation Equivalent operators:

$$\mathcal{G} = \mathcal{G}_L \circ \dots \mathcal{G}_\ell \circ \dots \circ \mathcal{G}_1, \quad \mathfrak{g} = \mathfrak{g}_L \circ \dots \mathfrak{g}_\ell \circ \dots \circ \mathfrak{g}_1, \; \mathfrak{g}_\ell = T^\dagger_{\Psi_{\ell+1}} \circ \mathcal{G} \circ T_{\Psi_\ell}$$

- $(\mathcal{G}, \mathfrak{g})$ are Representation Equivalent $\Rightarrow$ No Aliasing Error[1]
- Structure preserving discrete representations
- Natural Change of Frame formula:

$$\mathfrak{g}_\ell(\Psi'_\ell, \Psi'_{\ell+1}) = T^\dagger_{\Psi'_{\ell+1}} \circ T_{\Psi_{\ell+1}} \circ \mathfrak{g}_\ell(\Psi_\ell, \Psi_{\ell+1}) \circ T^\dagger_{\Psi_\ell} \circ T_{\Psi'_{\ell+1}}$$

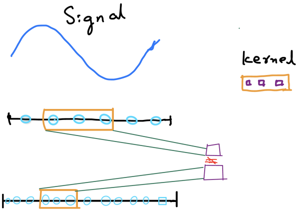
[1] Bounds on Aliasing Error $\Rightarrow$ Approximate ReNO

CNNs are not ReNOs !

- CNNs rely on Discrete Convolutions with fixed Kernel:

$$K_c[m] = \sum_{i=-s}^{s} k_i c[m-i]$$

- Pointwise evaluations with Sinc basis



- Easy to check that CNNs are Resolution dependent:

$$T_\Psi \circ K_\Psi(c) \circ T_\Psi^\dagger \neq T_{\Psi'} \circ K_{\Psi'}(d) \circ T_{\Psi'}^\dagger$$

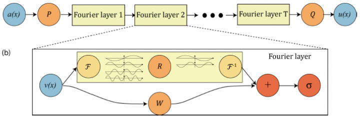
Alternative: Neural Operators of Kovachki et al, 2021

- Recall: DNNs are $\mathcal{L}_\theta = \sigma_K \odot \sigma_{K-1} \odot \dots\dots \sigma_1$
- Single hidden layer: $\sigma_k(y) = \sigma(A_k y + B_k)$
- Neural Operators generalize DNNs to ∞-dimensions:
- NO: $\mathcal{N}_\theta = \mathcal{N}_L \odot \mathcal{N}_{L-1} \odot \dots\dots \mathcal{N}_1$
- Single hidden layer;

$$(\mathcal{N}_\ell v)(x) = \sigma\left(B_\ell(x) + \int\limits_D K_\ell(x,y)v(y)dy\right)$$

- Kernel Integral Operators
- Different Kernels $\Rightarrow$ Low-Rank NOs, Graph NOs, Multipole NOs, ...

Fourier Neural Operators: Li et al, 2020



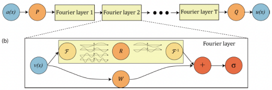
- Translation invariant Kernel $K(x, y) = K(x - y)$
- Use Fourier and Inverse Fourier Transform to define the KIO:

$$\int_D K_\ell(x, y) v(y) dy = \mathcal{F}^{-1}(\mathcal{F}(K)\mathcal{F}(v))(x)$$

- Parametrize Kernel in Fourier space.
- Fast implementation through FFT
- Excellent Empirical performance.
- Backed by solid Theory [2]

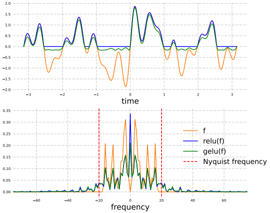
[2] Kovachki et al 2020, Lanthaler et al 2022

Are FNOs ReNOs ?



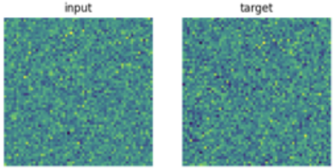
$$
\begin{tikzcd}
\mathcal{P}_K \arrow[r, "\mathcal{F}"] \arrow[d, "T^{\dagger}_{\Psi_K}"'] & \mathbb{C}^{2K+1} \arrow[r, "R\odot"] \arrow[d, leftrightarrow, "\mathrm{Id}"] & \mathbb{C}^{2K'+1} \arrow[r, "\mathcal{F}^{-1}"] \arrow[d, leftrightarrow, "\mathrm{Id}"] & \mathcal{P}_{K'} \\
\mathbb{C}^{2K+1} \arrow[r, "(2K+1)\cdot \mathrm{F}"] & \mathbb{C}^{2K+1} \arrow[r, "R\odot"] & \mathbb{C}^{2K'+1} \arrow[r, "\frac{1}{(2K'+1)}\cdot \mathrm{F}^{-1}"] & \mathbb{C}^{2K'+1} \arrow[u, "T_{\Psi_{K'}}"']
\end{tikzcd}
$$

- Convolution in Fourier space $\mathcal{K}$ + Nonlinearity σ
- $\mathcal{K}$ is ReNO wrt Periodic Bandlimited functions $\mathcal{P}_K$:
- Nonlinear activation σ can break bandlimits: $\sigma(f) \notin \mathcal{P}_K$
- FNOs are not necessarily ReNOs !!

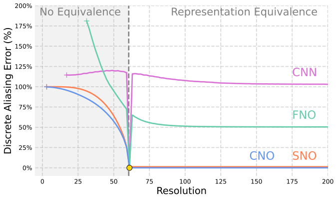


A Synthetic Example

- The underlying Operator:



- Errors:



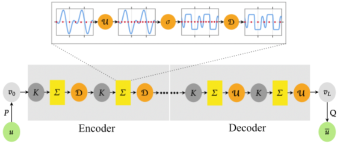
Can Convolutions be back in the reckoning ?

- Advantages of Convolution based models:
  - Variety of SOTA models in Vision etc.
  - Shift invariance.
  - Locality + Computational efficiency
  - CNNs closely linked with Finite difference Methods [3]
- Issue: Inconsistency in Function Space
- Plain vanilla CNNs and variants are not ReNOs

[3] Haber, Rutthoto, 2017

Convolutions Strike Back !!

- Convolutional Neural Operators: Raonic et al, Neurips 2023.

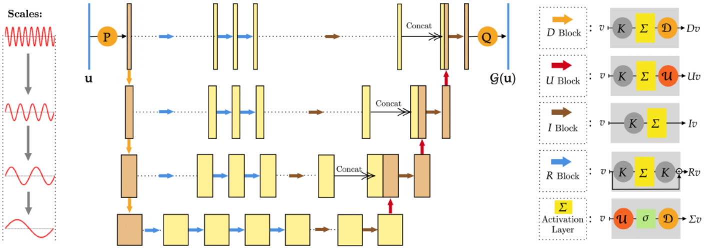


- Key Building Blocks [4]
- I: Use Continuous Convolutions on Bandlimited functions
  - Discrete locally supported convolution Kernels
- II: Modulated pointwise activations

$$
\begin{tikzcd}
\mathcal{B}_w \arrow[r, "\mathcal{K}_w"] & \mathcal{B}_w \arrow[d, "T^*_{\Psi_w}"] \\
\ell^2(\mathbb{Z}^2) \arrow[u, "T_{\Psi_w}"] \arrow[r] & \ell^2(\mathbb{Z}^2)
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{B}_w \arrow[r, hook, "P_{\mathcal{B}_{\overline{w}}(\mathbb{R}^2)}"] & \mathcal{B}_{\overline{w}} \arrow[r, "\sigma"] \arrow[d, "T^*_{\Psi_{\overline{w}}}"] & \mathcal{B}_{\overline{w}} \arrow[r, "P_{\mathcal{B}_w(\mathbb{R}^2)}"] & \mathcal{B}_w \arrow[d, "T^*_{\Psi_w}"] \\
\ell^2(\mathbb{Z}^2) \arrow[u, "T_{\Psi_w}"] \arrow[r, "\mathcal{U}_{w,\overline{w}}"] & \ell^2(\mathbb{Z}^2) \arrow[r, "\sigma"] & \ell^2(\mathbb{Z}^2) \arrow[u, "T_{\Psi_{\overline{w}}}"] \arrow[r, "\mathcal{D}_{\overline{w},w}"] & \ell^2(\mathbb{Z}^2)
\end{tikzcd}
$$

[4] Inspired by StyleGAN3 for Image Generation of Karras et al, 2022

CNO Architecture in Practice



- CNO instantiated as a modified Operator UNet
- Built for multiscale information processing

CNO properties

- CNO is a ReNO by construction.
- Universal Approximation Theorem:
- CNOs approximate any Continuous + operators $\mathcal{G} : H^r \mapsto H^s$
- Proof relies on building $\mathcal{G} \approx \mathcal{G}^* : \mathcal{B}_w \mapsto \mathcal{B}_{w'}$

$$\begin{array}{ccc} \mathcal{B}_w & \xrightarrow{\mathcal{G}^*} & \mathcal{B}_{w'}, \\ \big\downarrow T^*_{\Psi_w} & & T_{\Psi_{w'}} \big\uparrow \\ \ell^2(\mathbb{Z}^2) & \xrightarrow{\mathfrak{g}_{\Psi_w, \Psi_{w'}}} & \ell^2(\mathbb{Z}^2) \end{array}$$

- Efficient PyTorch implementation with CUDA kernels.
- Code available on https://github.com/bogdanraonic3/ConvolutionalNeuralOperator.git

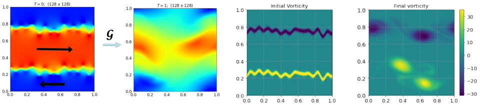
CNO Experiments

- Extensive Empirical evaluation on RPB benchmarks.

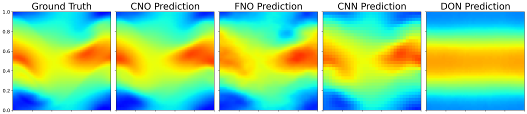
	In/Out	FFNN	GT	UNet	ResNet	DON	FNO	CNO
Poisson Equation	In	5.74%	2.77%	0.71%	0.43%	12.92%	4.98%	**0.21%**
	Out	5.35%	2.84%	1.27%	1.10%	9.15%	7.05%	**0.27%**
Wave Equation	In	2.51%	1,44%	1.51%	0.79%	2.26%	1.02%	**0.63%**
	Out	3.01%	1.79%	2.03%	1.36%	2.83%	1.77%	**1.17%**
Smooth Transport	In	7.09%	0.98%	0.49%	0.39%	1.14%	0.28%	**0.24%**
	Out	650.6%	875.4%	1.28%	0.96%	157.2%	3.90%	**0.46%**
Discontinuous Transport	In	13.0%	1.55%	1.31%	**1.01%**	5.78%	1.15%	**1.01%**
	Out	257.3%	22691.1%	1.35%	1.16%	117.1%	2.89%	**1.09%**
Allen-Cahn Equation	In	18.27%	0.77%	0.82%	1.40%	13.63%	**0.28%**	0.54%
	Out	46.93%	2.90%	2.18%	3.74%	19.86%	**1.10%**	2.23%
Navier-Stokes Equations	In	8.05%	4.14%	3.54%	3.69%	11.64%	3.57%	**2.76%**
	Out	16.12%	11.09%	10.93%	9.68%	15.05%	9.58%	**7.04%**
Darcy Flow	In	2.14%	0.86%	0.54%	0.42%	1.13%	0.80%	**0.38%**
	Out	2.23%	1.17%	0.64%	0.60%	1.61%	1.11%	**0.50%**
Compressible Euler	In	0.78%	2.09%	0.38%	1.70%	1.93%	0.44%	**0.35%**
	Out	1.34%	2.94%	0.76%	2.06%	2.88%	0.69%	**0.59%**

A Deep Dive: Navier-Stokes

- Operator:



- Comparison:

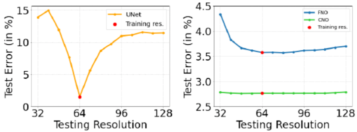


- Test Errors:

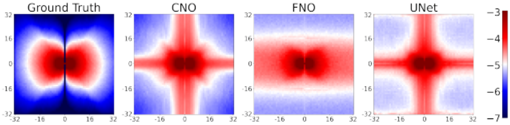
Model	FFNN	UNet	DeepONet	GT	FNO	CNO
Error	8.05%	3.54%	11.64%	4.02%	3.57%	2.76%

Further Results

- Resolution Dependence:

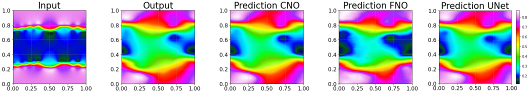


- Spectral Behavior: log spectra

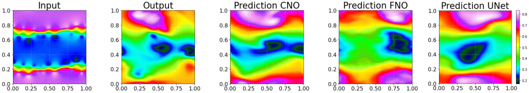


Out-of-Distribution Generalization or Zero-shot Learning

- Results for In-Distribution Testing:



- Results for Out-of-Distribution Testing:

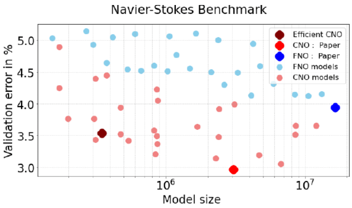


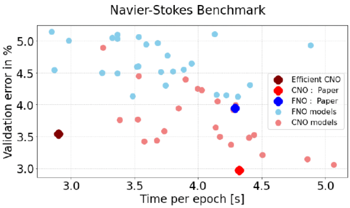
- Test Errors:

Model	FFNN	UNet	DeepONet	GTran	FNO	CNO
In	8.05%	3.54%	11.64%	4.02%	3.57%	2.76%
Out	16.12%	10.93%	15.05%	14.82%	9.58%	7.04%

- RunTime: 10^{-1}s on 100^2 grid for AzeBan vs 10^{-4}s for CNO

Computational Efficiency of CNO





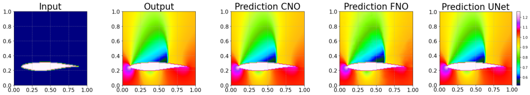
Similar Performance across the board !!

- Extensive Empirical evaluation on RPB benchmarks.

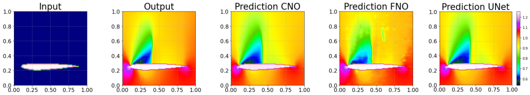
	In/Out	FFNN	GT	UNet	ResNet	DON	FNO	CNO
Poisson Equation	In	5.74%	2.77%	0.71%	0.43%	12.92%	4.98%	**0.21%**
	Out	5.35%	2.84%	1.27%	1.10%	9.15%	7.05%	**0.27%**
Wave Equation	In	2.51%	1,44%	1.51%	0.79%	2.26%	1.02%	**0.63%**
	Out	3.01%	1.79%	2.03%	1.36%	2.83%	1.77%	**1.17%**
Smooth Transport	In	7.09%	0.98%	0.49%	0.39%	1.14%	0.28%	**0.24%**
	Out	650.6%	875.4%	1.28%	0.96%	157.2%	3.90%	**0.46%**
Discontinuous Transport	In	13.0%	1.55%	1.31%	**1.01%**	5.78%	1.15%	**1.01%**
	Out	257.3%	22691.1%	1.35%	1.16%	117.1%	2.89%	**1.09%**
Allen-Cahn Equation	In	18.27%	0.77%	0.82%	1.40%	13.63%	**0.28%**	0.54%
	Out	46.93%	2.90%	2.18%	3.74%	19.86%	**1.10%**	2.23%
Navier-Stokes Equations	In	8.05%	4.14%	3.54%	3.69%	11.64%	3.57%	**2.76%**
	Out	16.12%	11.09%	10.93%	9.68%	15.05%	9.58%	**7.04%**
Darcy Flow	In	2.14%	0.86%	0.54%	0.42%	1.13%	0.80%	**0.38%**
	Out	2.23%	1.17%	0.64%	0.60%	1.61%	1.11%	**0.50%**
Compressible Euler	In	0.78%	2.09%	0.38%	1.70%	1.93%	0.44%	**0.35%**
	Out	1.34%	2.94%	0.76%	2.06%	2.88%	0.69%	**0.59%**

Compressible Euler Eqns

- Results for In-Distribution Testing:



- Results for Out-of-Distribution Testing:

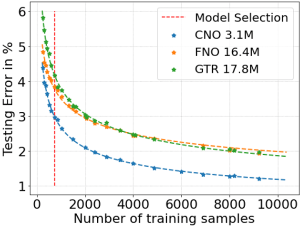


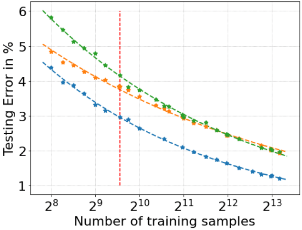
- Test Errors:

Model	FFNN	UNet	DeepONet	FNO	CNO
In	0.78%	0.38%	1.93%	0.44%	0.35%
Out	1.34%	0.76%	2.88%	0.69%	0.59%

- RunTime: 10^2s for NuwTun vs 10^{-4}s for CNO

Caveat: The need for Data



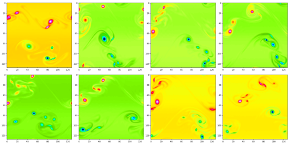


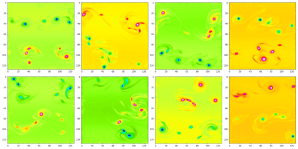
Possible Solutions

- Add Physics:
  - PINN type residual based loss functions
  - Forthcoming work on:
  - Preconditioned Physics-informed ReNOs
- Use data better through Foundation Models

Downstream Tasks

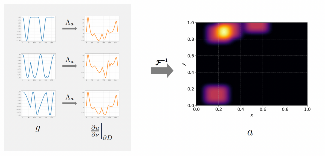
- PDE constrained optimization.
- Uncertainty Quantification.
  - Computation of Statistical Solutions for fluid flows.
  - Forthcoming work of (CamLab + Google Research)
  - CNO for low-frequencies.
  - Score-based Diffusion models for recovering high-frequencies

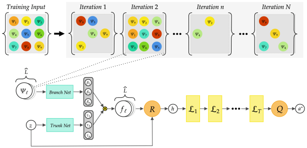




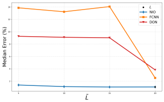
- Bayesian + Deterministic Inverse Problems.

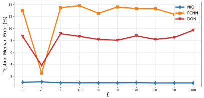
Neural Inverse Operators

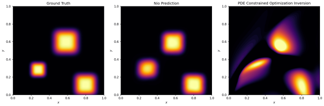




- Proposed in Molinaro, Yang, Engquist, SM ICML 2023.
- PDE Inverse Problems well-posed as Operators $\mapsto$ Functions
- NIO: Error 2.3% at 10^{-1}s vs PDE-Opt: 11.2% at 8.5hr







Summary

- CDE is essential for Operator Learning.
- Lack of CDE $\Rightarrow$ Aliasing Errors
- Developed CDE for Operators to define ReNO
- CNO as a convolution based ReNO.
- Competitive performance on RPB datasets.
- Extensions of CNO:
  - More test cases, 3D
  - Non-Cartesian Geometries.
  - Auto-regressive time marching
  - Limited Data needs Physics Informed ReNOs
  - Explicit Error Bounds
- Out-of-Generalization beyond Deformations or Symmetries
- Design of Foundation Models