HELMHOLTZAI | ARTIFICIAL INTELLIGENCE COOPERATION UNIT

Helmholtz-Zentrum hereon

Accurate Long Rollouts of Fluid Dynamics Achieved by Symmetrically-physically Constrained Neural PDE Solvers

Speaker: Dr. Yunfei Huang

Supervisor: Dr. David S. Greenberg

The group of Model-driven Machine Learning, Helmholtz-Zentrum Hereon

Outline

(1) Introduction

(2) Symmetry- and Physics-Constrained Neural Integrators

(3) Experiments

(4) Results

(5) Conclusion

Outline

(1) Introduction

(2) Symmetry- and Physics-Constrained Neural Integrators

(3) Experiments

(4) Results

(5) Conclusion

Neural PDE Integration

A general PDE: $$\frac{\partial \boldsymbol{w}}{\partial t} = \mathcal{F}(t, \boldsymbol{x}, \boldsymbol{w}, \nabla \boldsymbol{w}, \nabla^2 \boldsymbol{w}, \ldots)$$

space and time
$x \in \Omega \subset \mathbb{R}^d$
$t \in [0, T]$

$\boldsymbol{w}(t, x) \in \mathbb{R}^m$, initial and boundary conditions: $B[\boldsymbol{w}](t, \boldsymbol{x}) = 0, \forall x \in \partial\Omega$

An update operator: $$\boldsymbol{w}(t + \Delta t, \cdot) = \mathcal{G}[\boldsymbol{w}(t, \cdot)]$$

Traditional: Numerical solver

Recent: Trained neural networks (called neural PDE integration)

Neural PDE Integration

Why is neural PDE integration:

Makes computing fast: coarse resolution and multiple time steps

Especially important for climate and ocean systems

Challenges of neural solver :

to obtain an accurate long-stable rollout

generalization outside of trained data

Our scientific question:

Could we benefit from the physical and symmetry constraints for accurate long-stable rollout?

Outline

(1) Introduction

(2) Symmetry- and Physics-Constrained Neural Integrators

(3) Experiments

(4) Results

(5) Conclusion

Outline

(1) Introduction

(2) Symmetry- and Physics-Constrained Neural Integrators

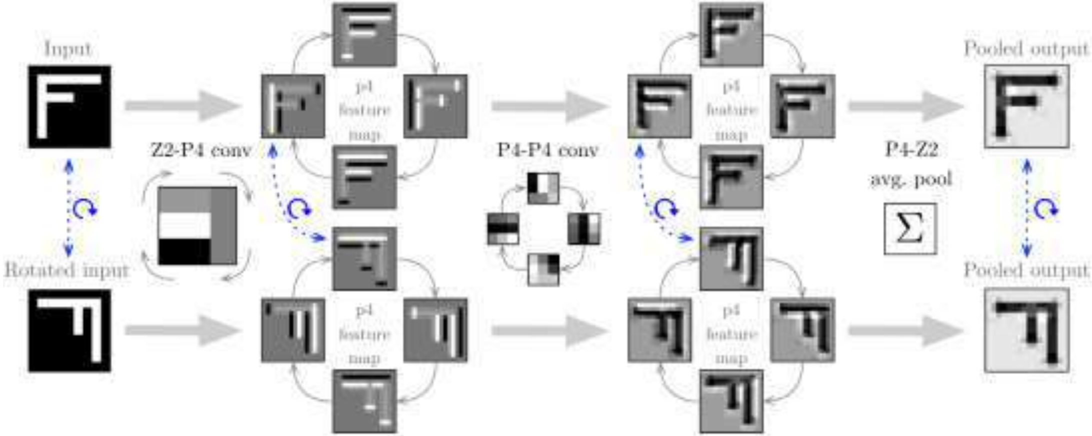
(3) Experiments

(4) Results

(5) Conclusion

Symmetry- and Physics-Constrained Neural Integrators

Rotation Equivariant CNNs



Cohen et al. 2016
Veeling et al. 2018

$$t(g(x)) = g(t(x))$$

$t(\cdot)$: transformations $g(\cdot)$: networks

New fast versions

General E(2)-Equivariant Steerable CNNs

Jenner et al. 2022

E(n)-equivariant Steerable CNNs (escnn)

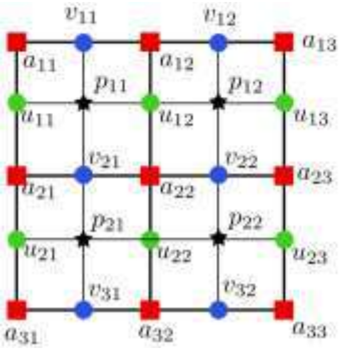
Cesa et al. 2022

Symmetry- and Physics-Constrained Neural Integrators

Symmetries of incompressible Navier–Stokes equations

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\frac{\nabla p}{\rho} + \mu \nabla^2 \boldsymbol{u} + \boldsymbol{f}; \qquad \nabla \cdot \boldsymbol{u} = 0$$

C-grid staggering

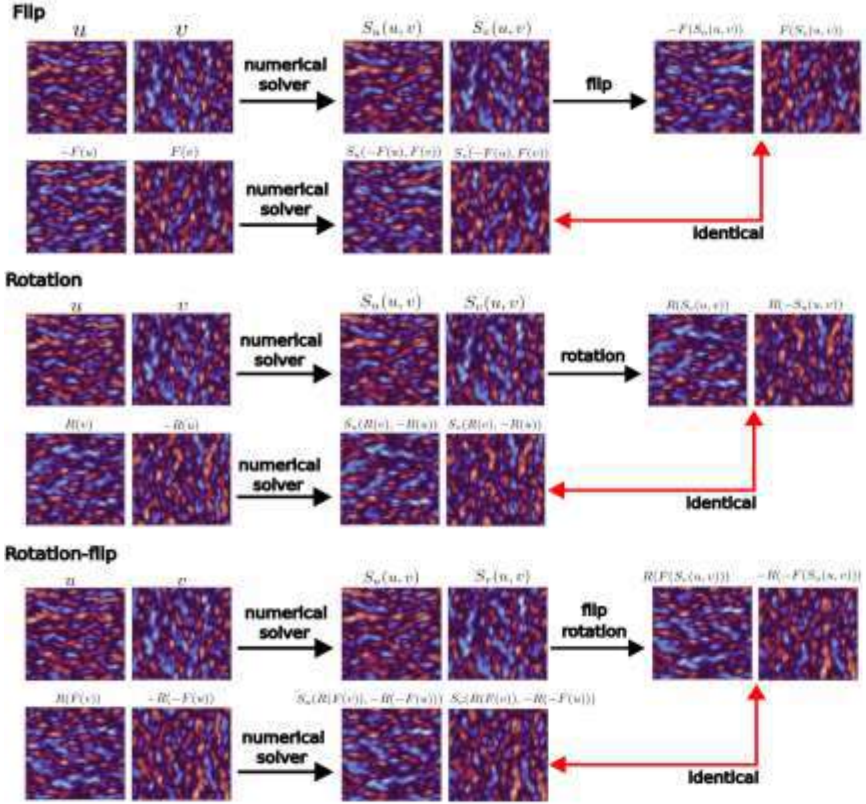


Symmetries of INS

$$\text{flip}: \begin{cases} S_u(-F(u), F(v)) = -F(S_u(u,v)) \\ S_v(-F(u), F(v)) = F(S_v(u,v)) \end{cases}$$

$$\text{rotation}: \begin{cases} S_u(R(v), -R(u)) = R(S_v(u,v)) \\ S_v(R(v), -R(u)) = R(-S_u(u,v)) \end{cases}$$

$$\text{flip}-\text{rotation}: \begin{cases} S_u(R(F(v)), -R(-F(u))) = R(F(S_v(u,v))) \\ S_v(R(F(v)), -R(-F(u))) = -R(-F(S_u(u,v))) \end{cases}$$



Symmetry- and Physics-Constrained Neural Integrators

We build group equivariant (p4, p4m) input layers for INS

According to the symmetries of INS, we create an input layer for p4.

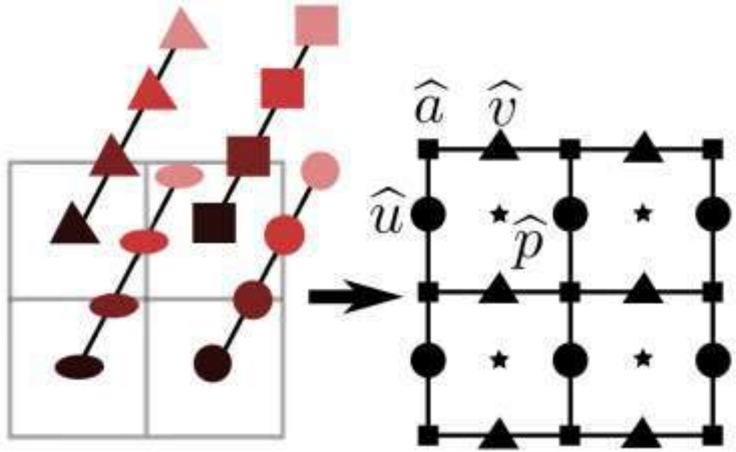
$$
\begin{aligned}
y^1_{j,0,\cdot,\cdot} &= \sum_{i=0}^{c^u_{in}-1} \left(W^u_{j,i,\cdot,\cdot} \star u_{i,\cdot,\cdot}\right) + \sum_{i=0}^{c^v_{in}-1} \left(W^v_{j,i,\cdot,\cdot} \star v_{i,\cdot,\cdot}\right) + b_j, \\
y^1_{j,1,\cdot,\cdot} &= \sum_{i=0}^{c^u_{in}-1} \left(R^{90^\circ}_{\text{rot}}\left(W^v_{j,i,\cdot,\cdot}\right) \star u_{i,\cdot,\cdot}\right) + \sum_{i=0}^{c^v_{in}-1} \left(-R^{90^\circ}_{\text{rot}}\left(W^u_{j,i,\cdot,\cdot}\right) \star v_{i,\cdot,\cdot}\right) + b_j, \\
y^1_{j,2,\cdot,\cdot} &= \sum_{i=0}^{c^u_{in}-1} \left(-R^{180^\circ}_{\text{rot}}\left(W^u_{j,i,\cdot,\cdot}\right) \star u_{i,\cdot,\cdot}\right) + \sum_{i=0}^{c^v_{in}-1} \left(-R^{180^\circ}_{\text{rot}}\left(W^v_{j,i,\cdot,\cdot}\right) \star v_{i,\cdot,\cdot}\right) + b_j, \\
y^1_{j,3,\cdot,\cdot} &= \sum_{i=0}^{c^u_{in}-1} \left(-R^{270^\circ}_{\text{rot}}\left(W^v_{j,i,\cdot,\cdot}\right) \star u_{i,\cdot,\cdot}\right) + \sum_{i=0}^{c^v_{in}-1} \left(R^{270^\circ}_{\text{rot}}\left(W^u_{j,i,\cdot,\cdot}\right) \star v_{i,\cdot,\cdot}\right) + b_j.
\end{aligned}
$$

Symmetry- and Physics-Constrained Neural Integrators

We build vector output layers on C-grid

p4 $u_{i+0.5,j} = p_{i+1,j,0} - p_{i,j,1}$

$v_{i,j+0.5} = p_{i,j+1,2} - p_{i,j,3}$



p4m $u_{i+0.5,j} = p_{i+1,j,1} - p_{i,j,3} + p_{i+1,j,5} - p_{i,j,7}$

$v_{i,j+0.5} = p_{i,j+1,2} - p_{i,j,4} + p_{i,j+1,6} - p_{i,j,0}$

Symmetry- and Physics-Constrained Neural Integrators

Physical constraints embedded into networks

Momentum conservation

$$u^{t+1} = u^t + du - \mathrm{mean}(du)$$
$$v^{t+1} = v^t + dv - \mathrm{mean}(dv)$$

Helmholtz decomposition $\vec{v} = \nabla q + \nabla \times \vec{a}$

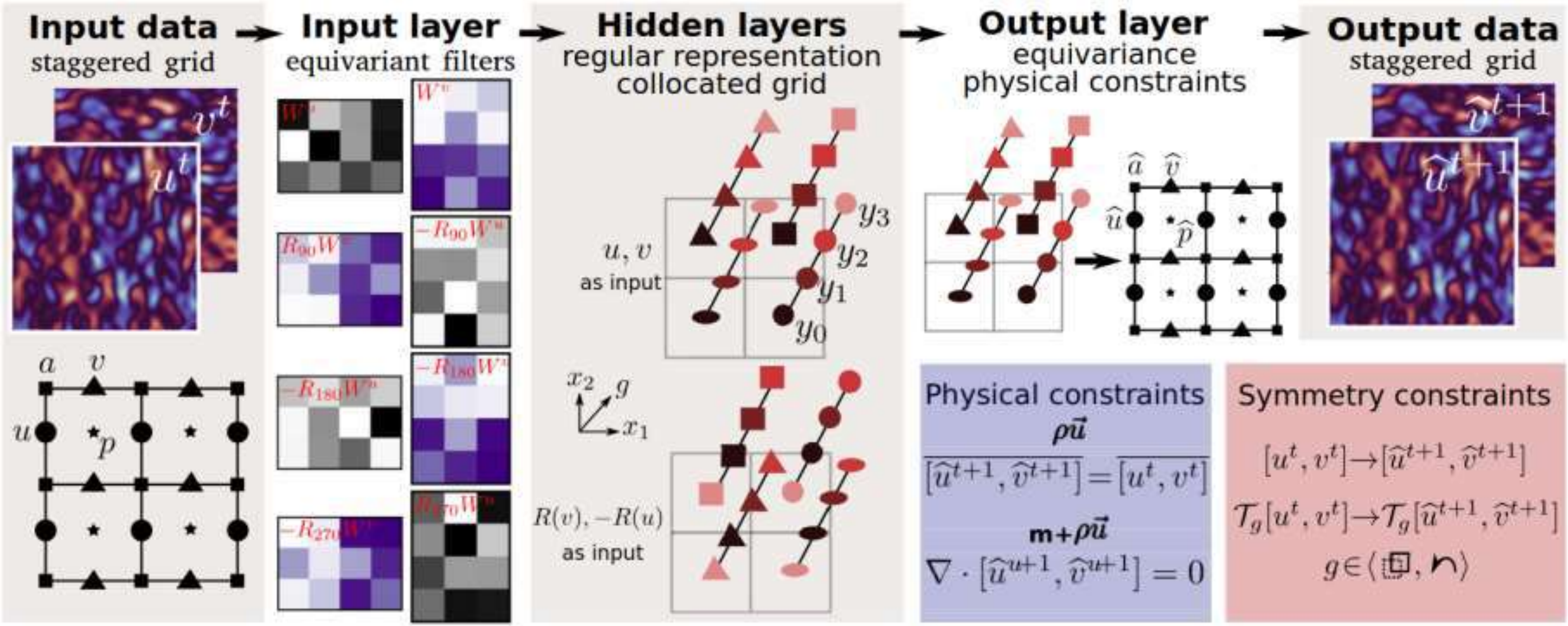
Learn scalar potential

$$u^{t+1} = u^t - \frac{\partial a}{\partial y}$$
$$v^{t+1} = v^t + \frac{\partial a}{\partial x}$$

Wandel et al. 2020

Symmetry- and Physics-Constrained Neural Integrators

The symmetry- and physics-constrained neural integrator for INS



Outline

Outline

(1) Introduction

(2) Symmetry- and Physics-Constrained Neural Integrators

(3) Experiments

(4) Results

(5) Conclusion

Experiments

Shallow water equations

$$\frac{\partial \vec{u}}{\partial t} = -C_D \frac{1}{h} \vec{u}|\vec{u}| - g\nabla\zeta + a_h \nabla^2 \vec{u}$$

$$\frac{\partial \zeta}{\partial t} = -\nabla \cdot (h\vec{u})$$

Incompressible Navier–Stokes equations

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\frac{\nabla p}{\rho} + \mu \nabla^2 \boldsymbol{u} + \boldsymbol{f}$$

$$\nabla \cdot \boldsymbol{u} = 0$$

Table 1: Geometric and physical constraints for SWEs

	Symmetries		
Conservation laws			
None ∅	**p1/∅**	**p4/∅**	**p4m/∅**
Mass **m**	**p1/m**	**p4/m**	**p4m/m**

Table 2: Geometric and physical constraints for INS

	Symmetries		
Conservation laws			
None ∅	**p1/∅**	**p4/∅**	**p4m/∅**
Momentum $\rho\vec{u}$	**p1/**$\rho\vec{u}$	**p4/**$\rho\vec{u}$	**p4m/**$\rho\vec{u}$
Mass/moment. **m+**$\rho\vec{u}$	**p1/m+**$\rho\vec{u}$	**p4/m+**$\rho\vec{u}$	**p4m/m+**$\rho\vec{u}$

Outline

(1) Introduction

(2) Symmetry- and Physics-Constrained Neural Integrators

(3) Experiments

(4) Results

(5) Conclusion

Outline

(1) Introduction

(2) Symmetry- and Physics-Constrained Neural Integrators

(3) Experiments

(4) Results

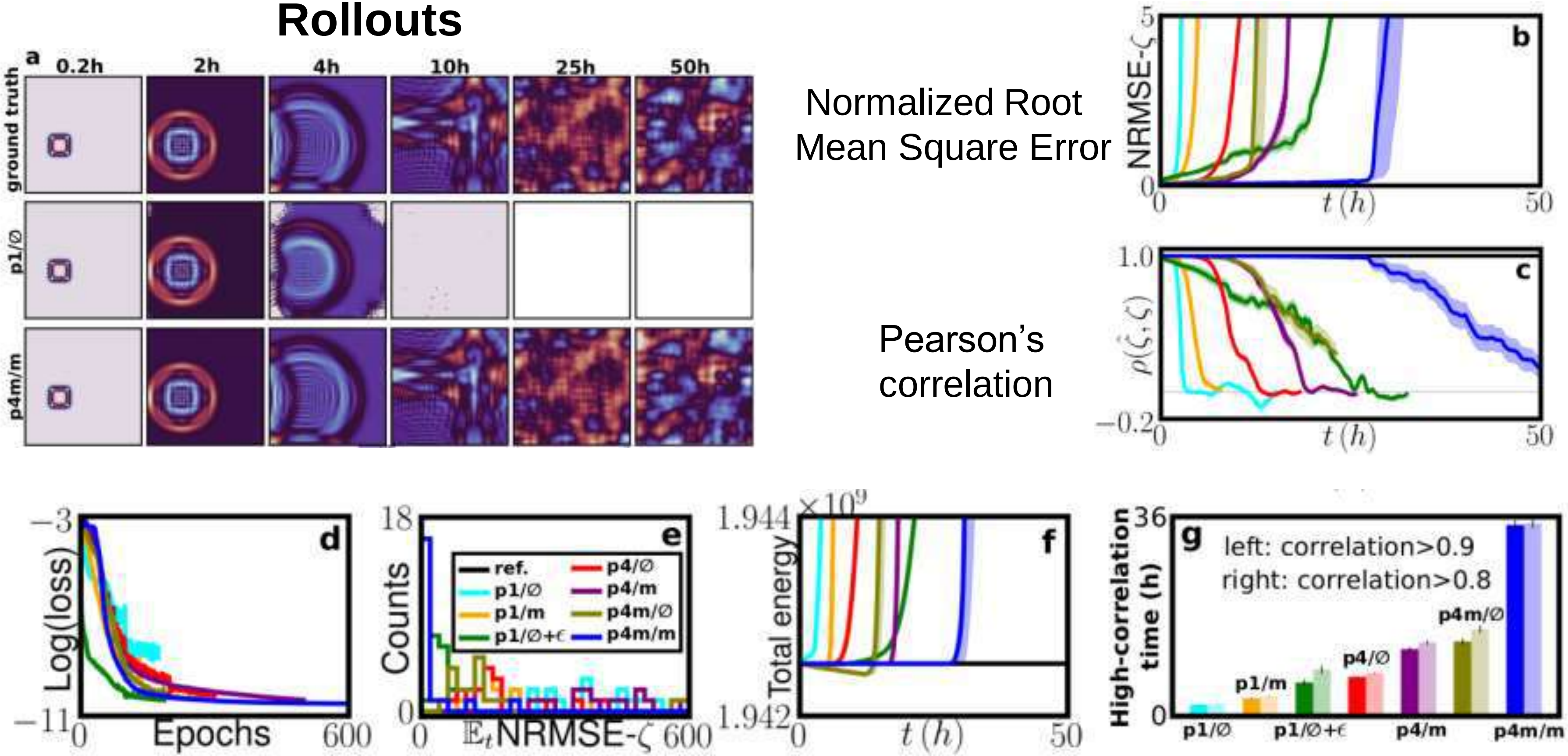
(5) Conclusion

P4m/m outperforms other networks for SWE

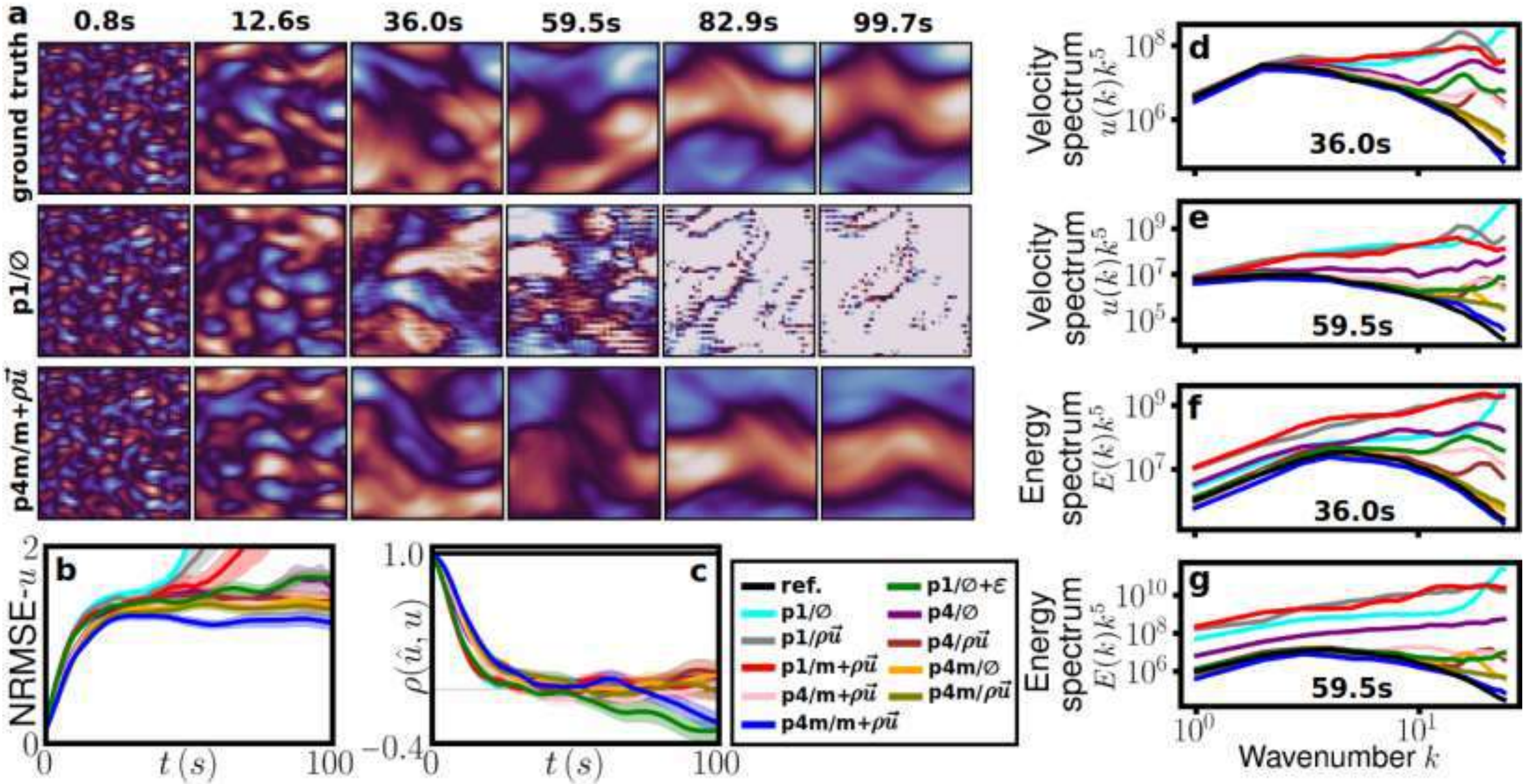
Rollouts

Normalized Root Mean Square Error

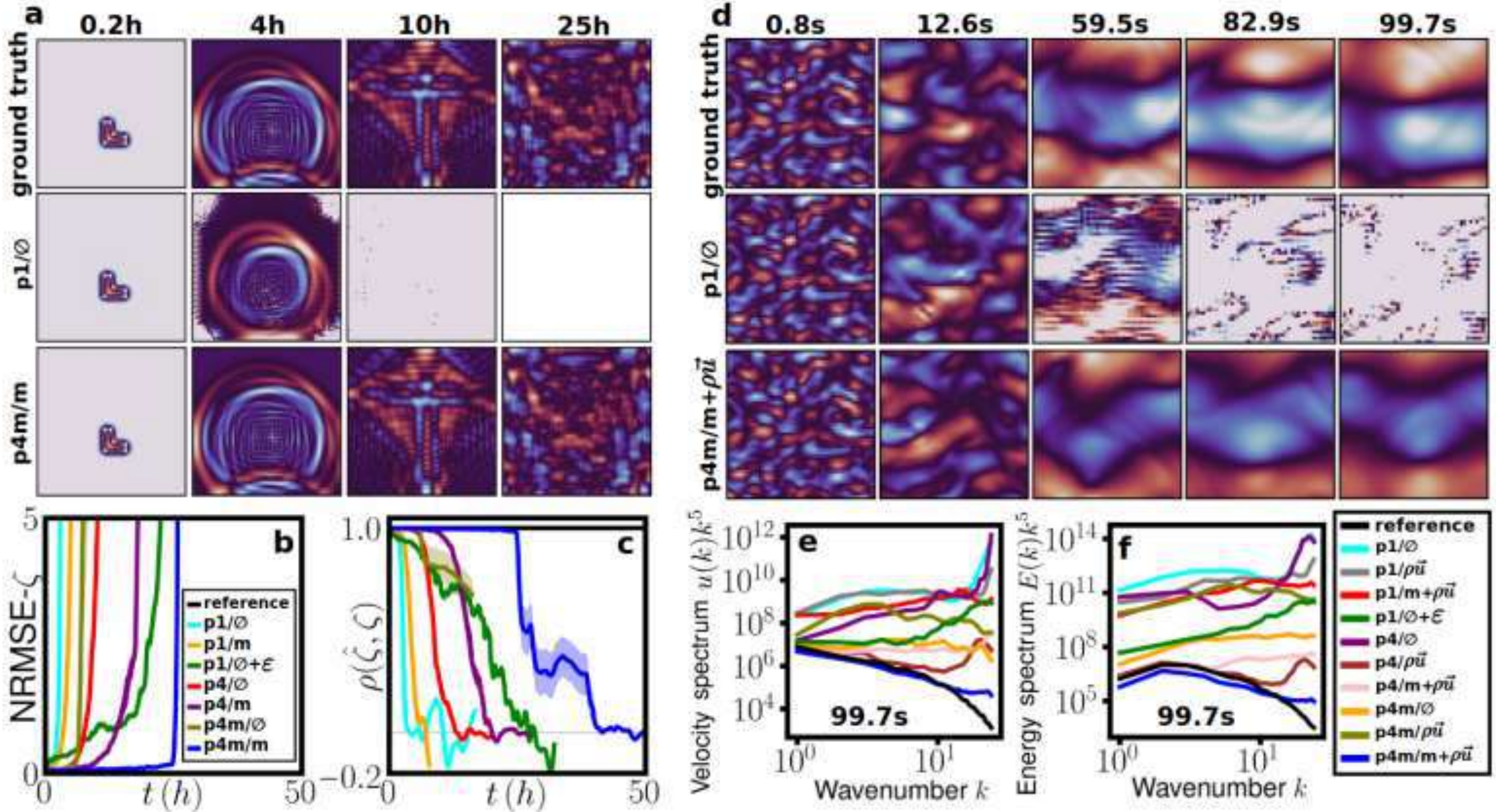
Pearson's correlation



P4m/m+ρ⃗ achieves a long rollout for decaying turbulence

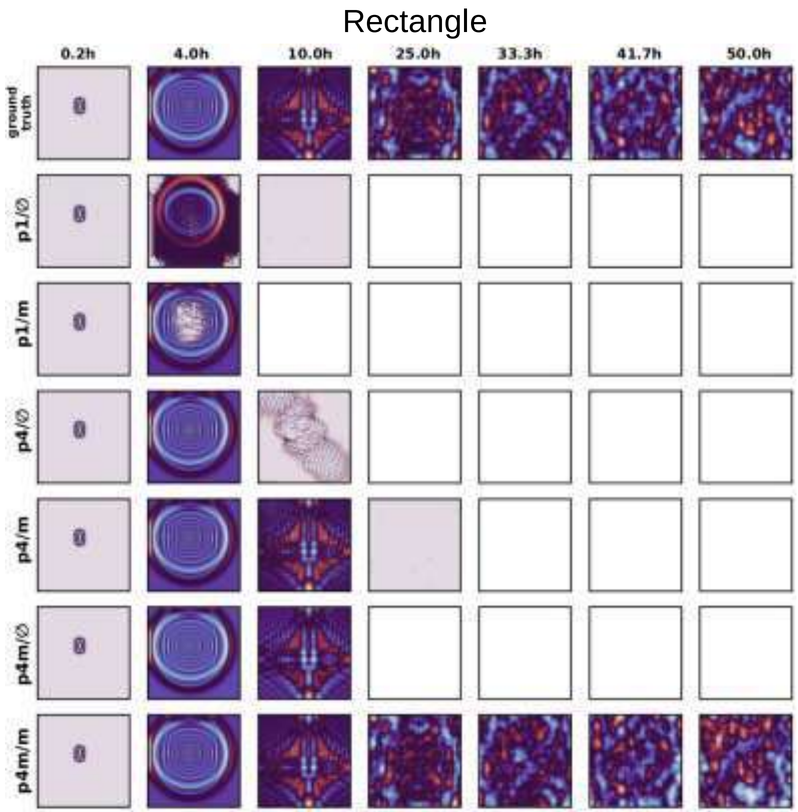


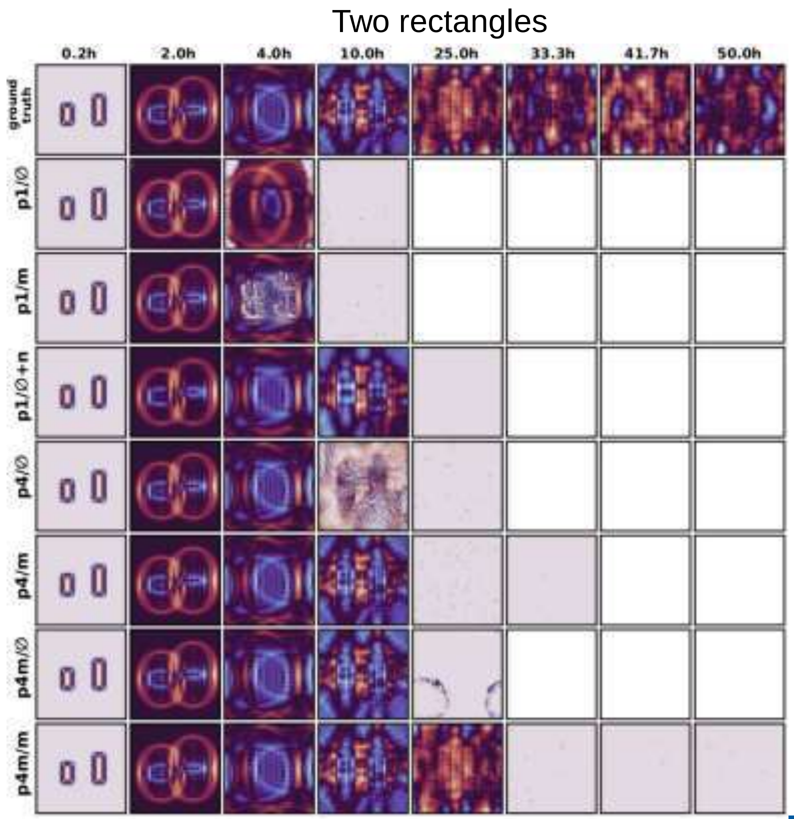
Generalization beyond the training distributions



More generalizations for SWEs

Training is one square as initial condition.



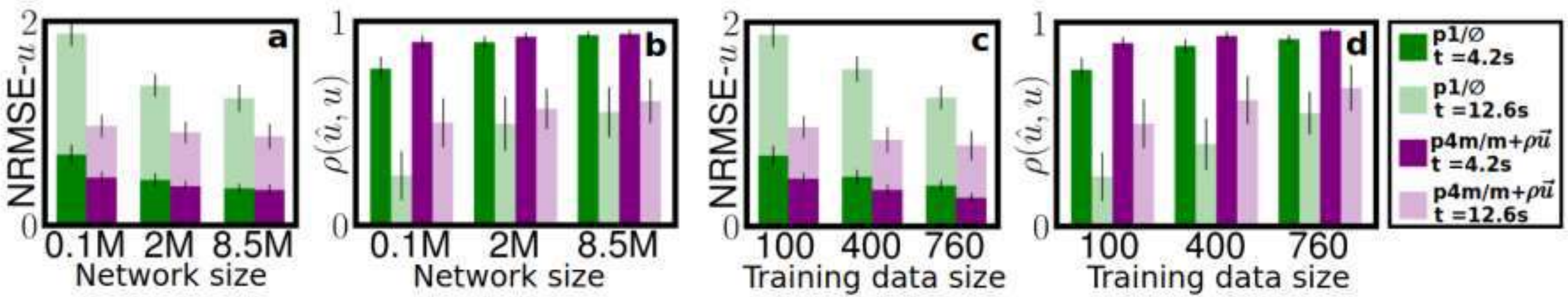


Stability of symmetry-and physics-constrained model

Using incompressible Navier–Stokes equations for test

Different network size

Different data size



Outline

(1) Introduction

(2) Symmetry- and Physics-Constrained Neural Integrators

(3) Experiments

(4) Results

(5) Conclusion

Outline

(1) Introduction

(2) Symmetry- and Physics-Constrained Neural Integrators

(3) Experiments

(4) Results

(5) Conclusion

Conclusion

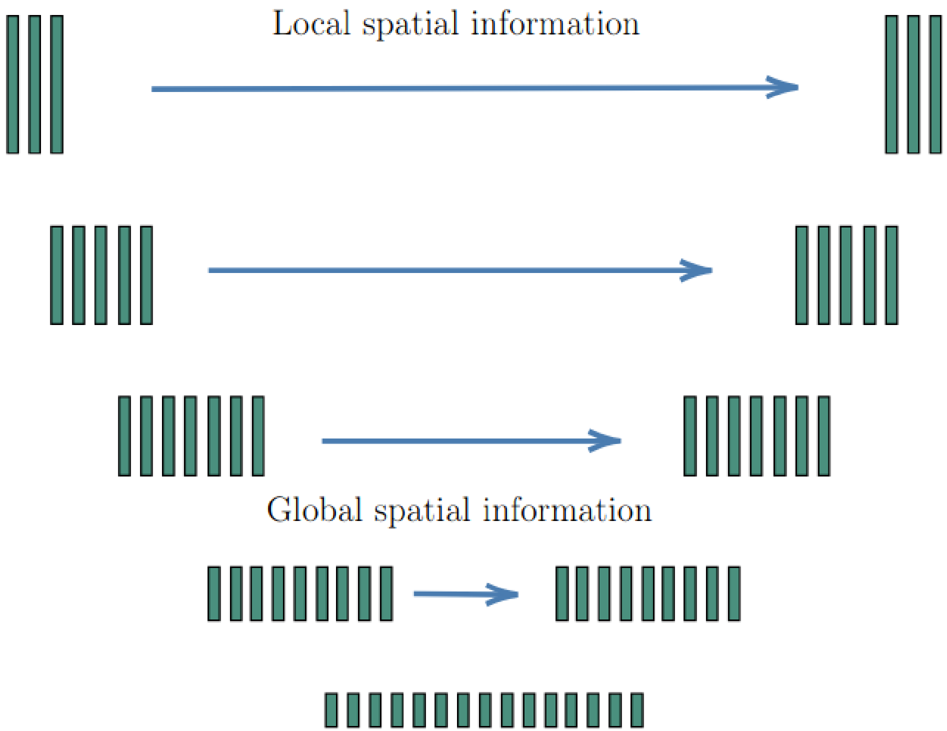
we present a novel architectural component that simultaneously enforces hard constraints on symmetries and conservation laws for neural PDE integrators

we find that greater enhancements are brought about by symmetry constraints, but these do not make conservation laws redundant.

Maximally constrained networks provided the best match to reference simulations, both for individual initial conditions and on a statistical level.

Thank you for your attention

Architecture



modern U-Net architectures

Gupta et al., 2022;
Ho et al., 2020;
Nichol et al., 2021;
Ramesh et al., 2021;