Exploring Conditioning Strategies for Fourier Neural Operators

Improving PDE Forecasting with Parameter-based Models

Wenxi Cai, Yimin Wang

caiwenxi@umich.edu, wyimin@umich.edu

University of Michigan

Motivation & Challenges

The Challenge with PDEs

Forecasting physical systems (fluids, waves) is computationally expensive with traditional solvers. Real-world applications often require re-evaluating systems under varying physical parameters (e.g., viscosity, wave speed), which is prohibitive for real-time use.

Limitations of the FNO

While Fourier Neural Operators (FNO) are promising, they often struggle with:

- **Parameter Injection:** Standard FNOs lack an explicit mechanism for incorporating the physical parameters
- **Error Accumulation:** Drift over long rollout horizons.

Problem Definition

- **Main task:** We study PDEs whose behavior changes with physical parameters $\mathcal{F}(u; p) = 0$
- **Our goal:** Establish a FNO that combine the physical parameters as a condition to predict the u at the next time step based on u_0: $\mathcal{G}_\theta$: $(u_0, p) \mapsto \hat{u}$
- **Minor task:** Reinforce the accuracy of one-step prediction, and then do a long horizontal rollout.

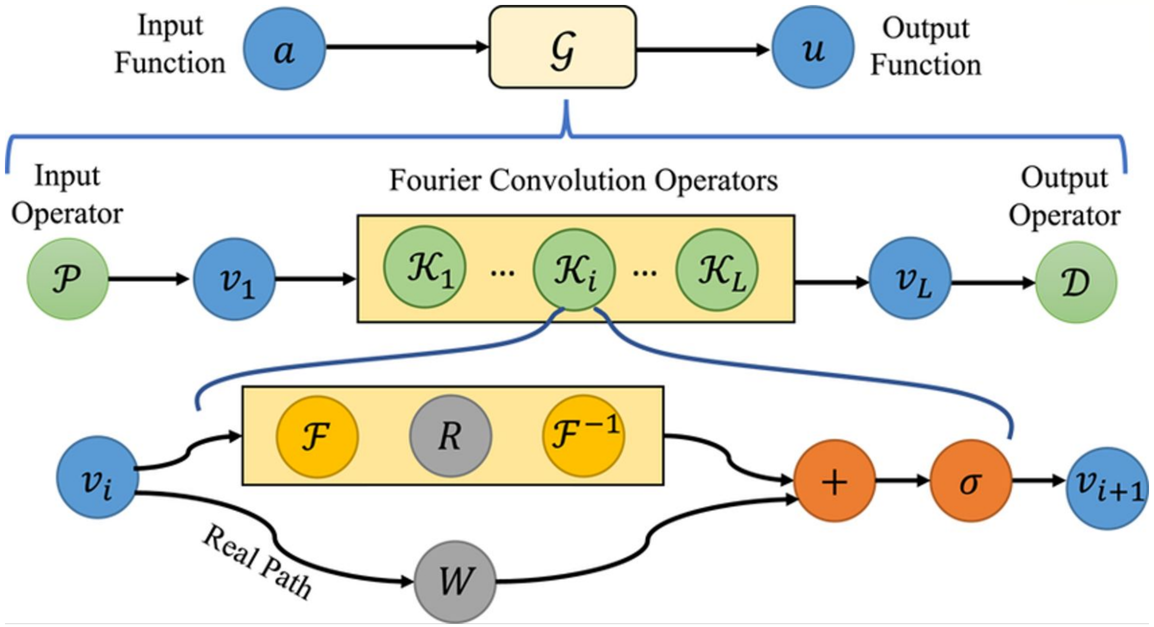
$$\hat{u}(t_1) = \mathcal{G}_\theta(u_0, p), \hat{u}(t_2) = \mathcal{G}_\theta(\hat{u}(t_1), p), \dots, \hat{u}(t_K) = \mathcal{G}_\theta(\hat{u}(t_{K-1}), p)$$

- **Error accumulation:** Suppose the error is $\frac{|\mathcal{G}_\theta(u(t_k),p) - u(t_{k+1})|}{|u(t_{k+1})|} \approx a\%$, after n rollout steps, the error will be $(1 + a\%)^n$

Fourier Neural Operator (FNO)

The FNO learns mappings between infinite dimensional function spaces.

- **Spectral Convolution:** Operates in the frequency domain, capturing global dependencies efficiently.
- **Resolution Invariant:** Can be evaluated at different grid resolutions.
- **The Gap:** However, standard FNO architectures have no clear mechanism for integrating physical parameters. These parameters significantly influence solution structure.



Ref:https://www.researchgate.net/figure/Flowchart-of-the-Fourier-neural-operator-FNO-approach-At-the-top-of-the-figure-a-is_fig1_381490671

How should physical parameters be incorporated into FNO?

Three Conditioning Mechanisms

1. Local Conditioning

Method: Feature-wise modulation between two spectral blocks.

Effect: Allows parameters to locally scale and shift intermediate features frequency-by-frequency.

2. Global Conditioning

Method: Channel-wise scaling after the operator.

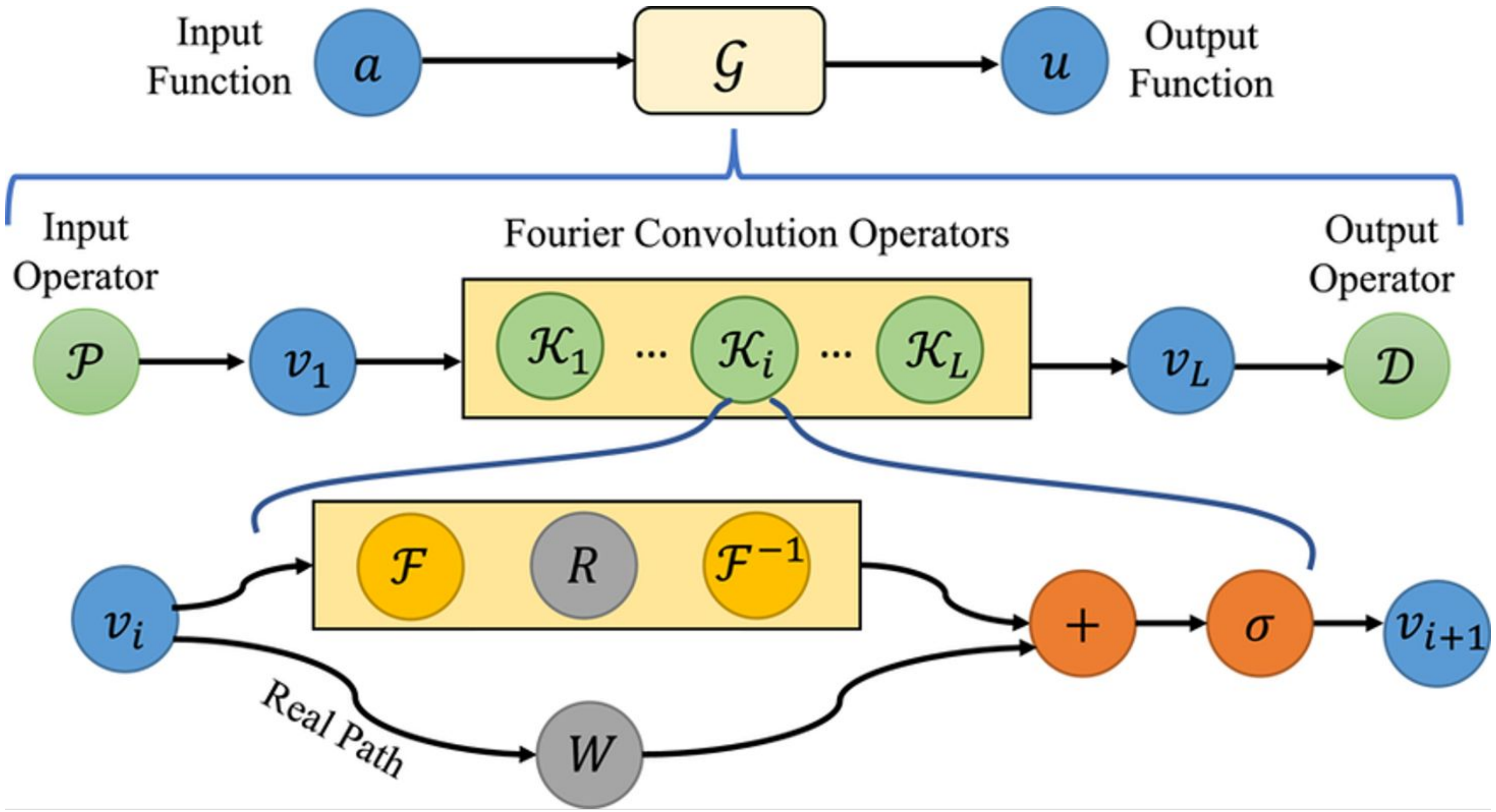
Effect: Similar to DeepONet trunk/branch. Adjusts the global amplitude of learned features without modifying spatial patterns.

3. Input Concatenation

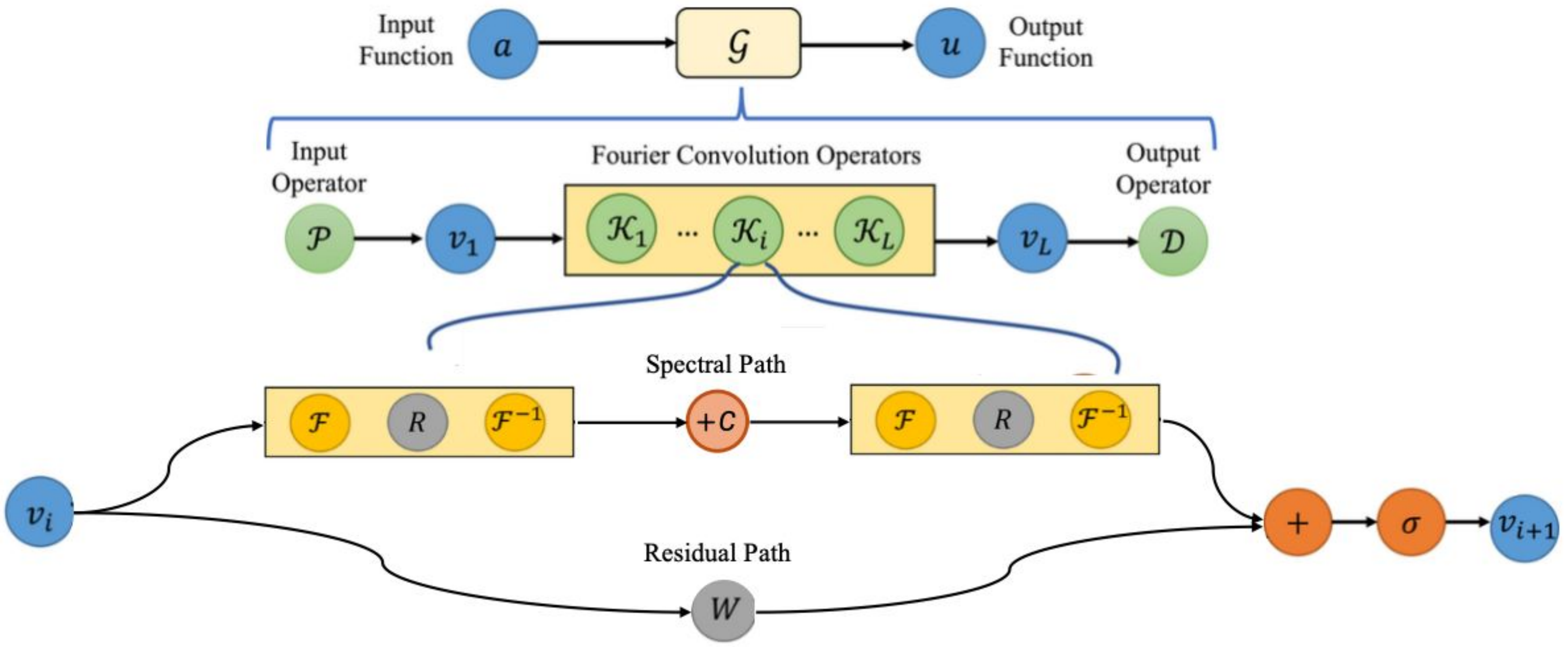
Method: Appending parameters as extra channels.

Effect: Rely on the network to implicitly learn the parameter interactions.

Baseline: FNO

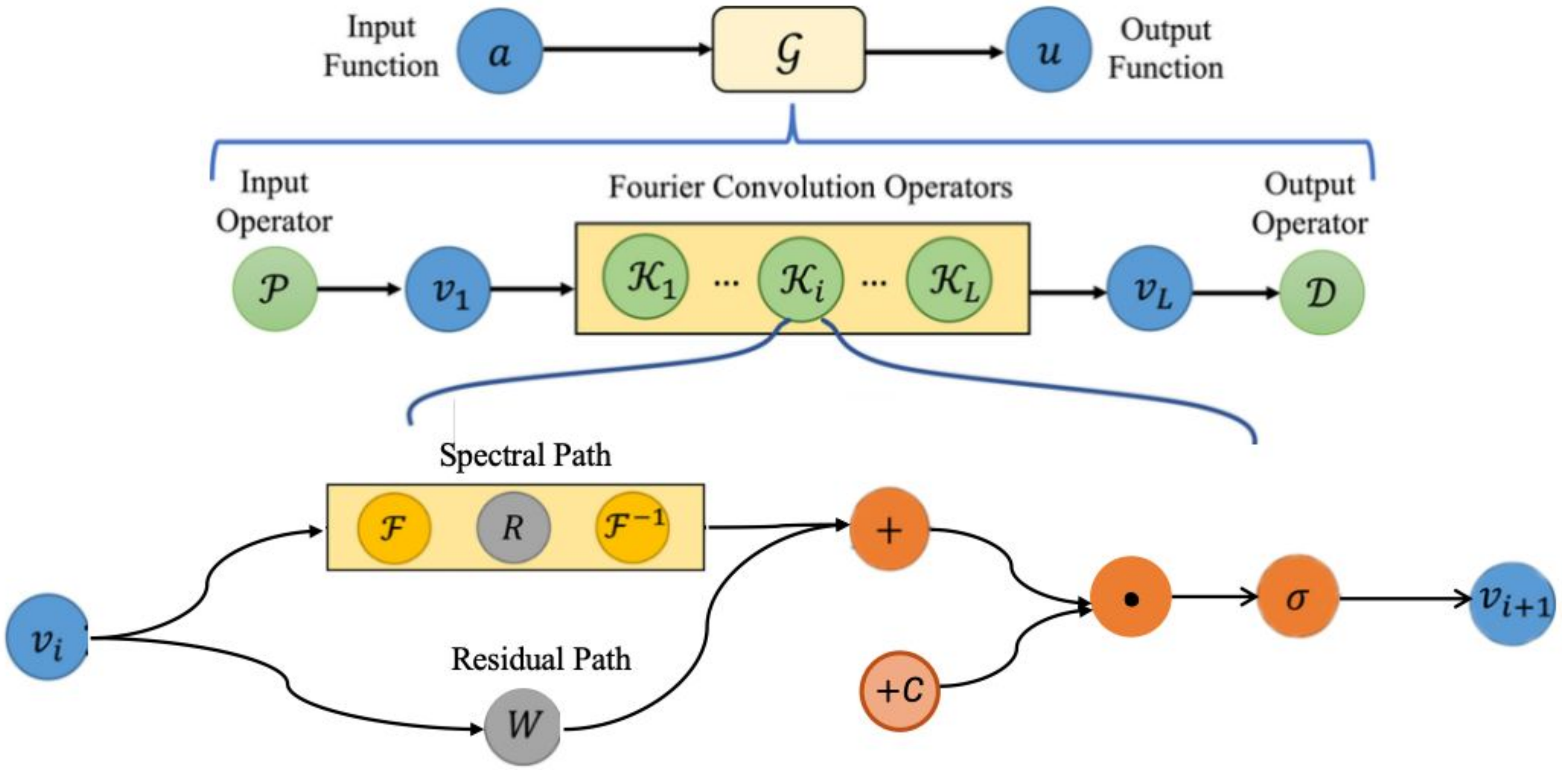


Local Conditioning



Equivalent to $y = x * f_{|\mathrm{p}} + b_{|\mathrm{p}}$ for spectral path.

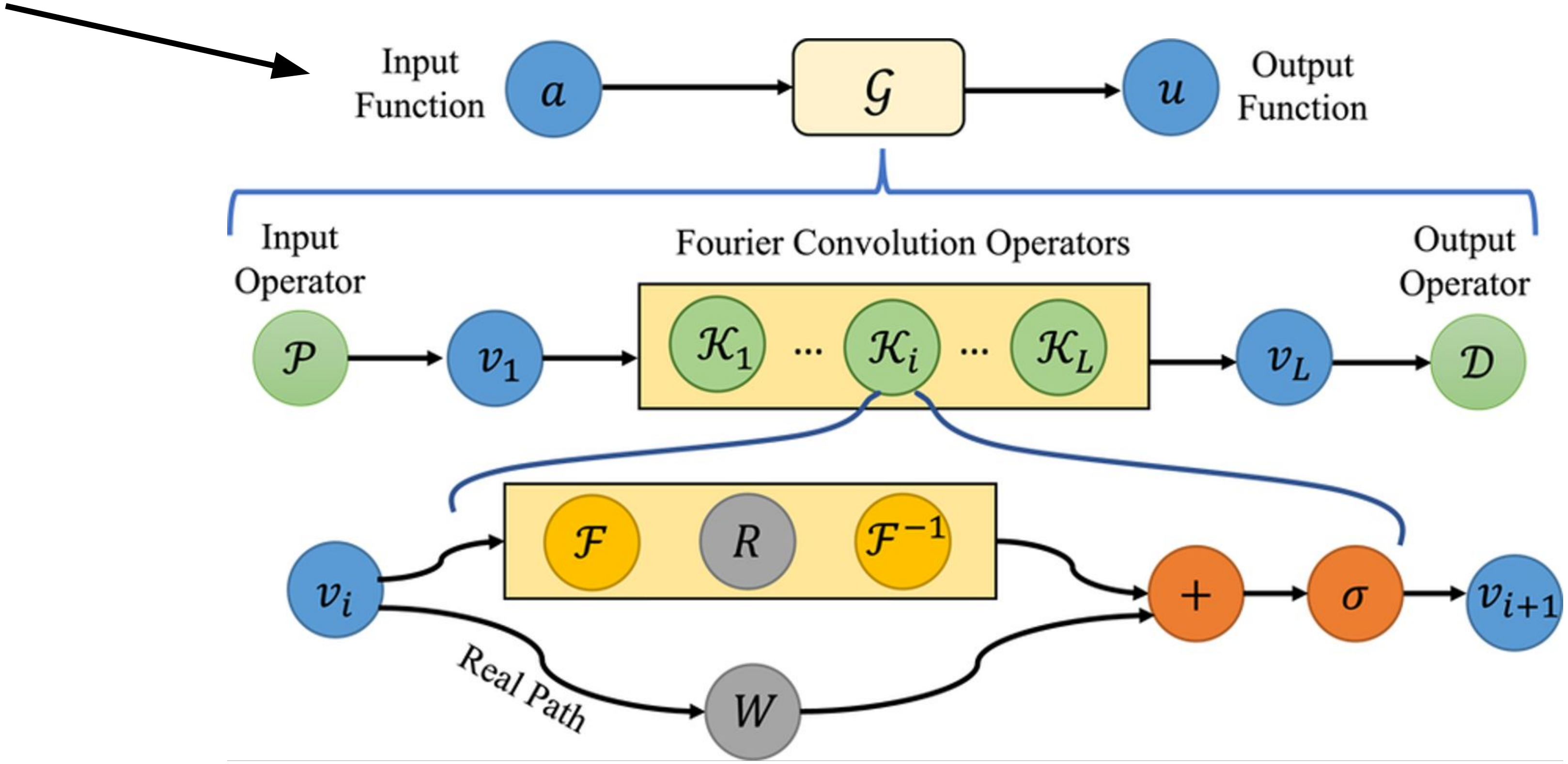
Global Conditioning



Equivalent to $y = x * f|_{\mathrm{p}}$ **for spectral path.**

Input Concatenation

Concatenation of input u0 and broadcasted condition p



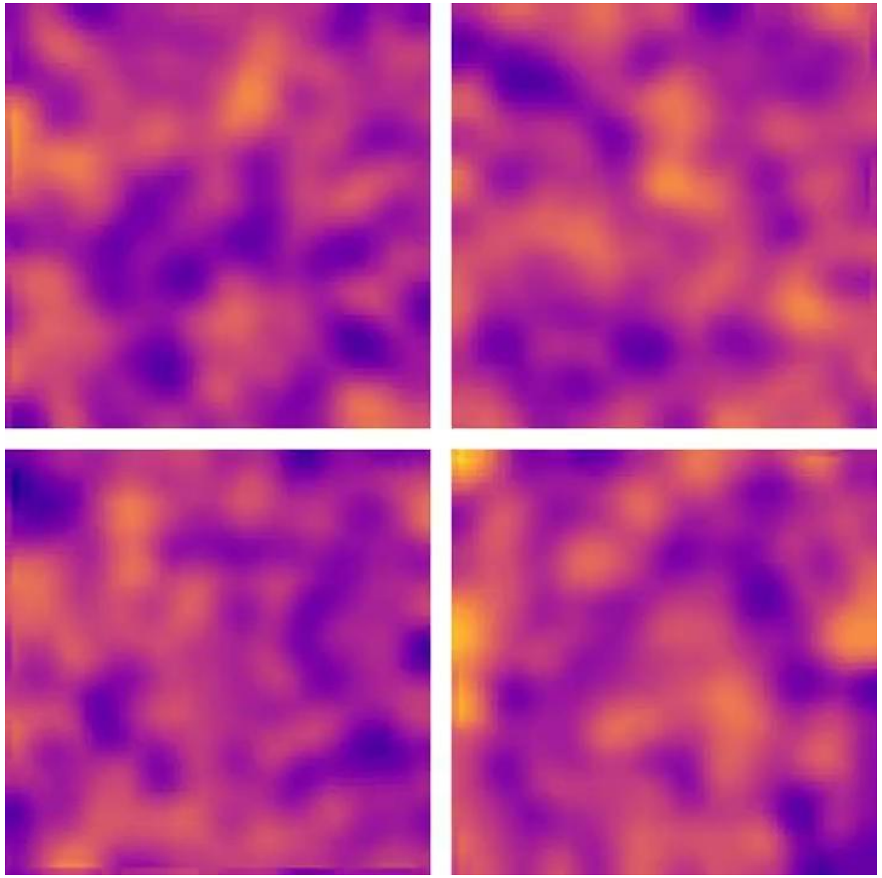
Learn the condition implicitly.

Navier-Stokes

Varying forcing coefficients

Experimental Benchmarks

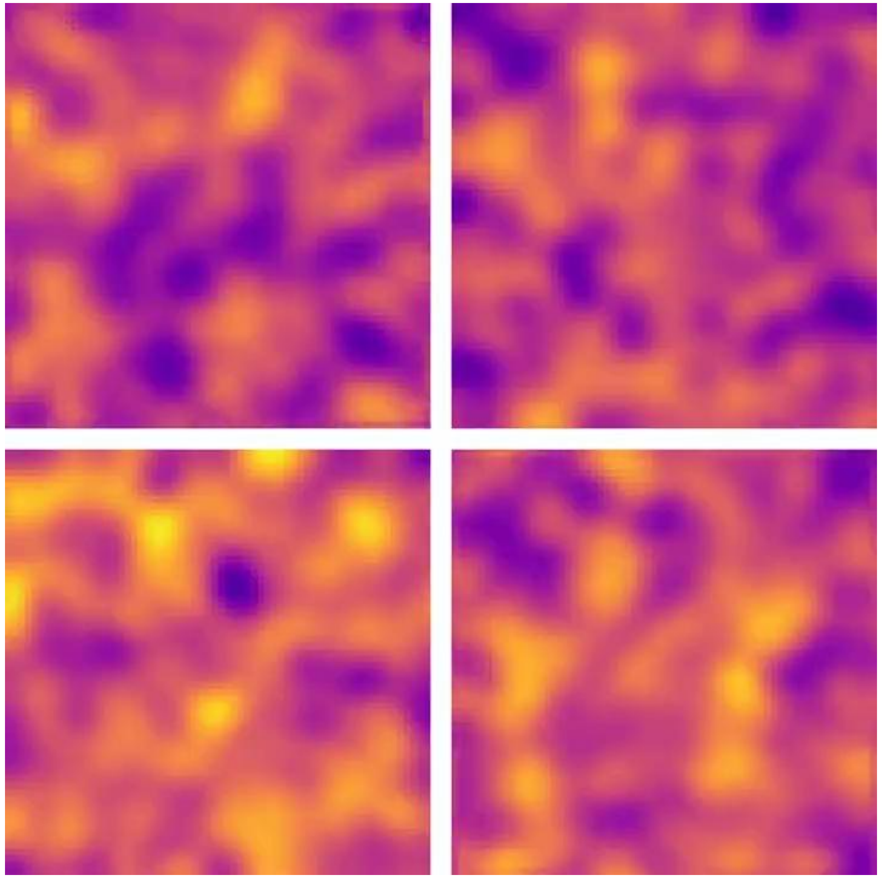
$$\partial_t u = \nu \Delta u - v \cdot \nabla u + s(x, y, t)$$



Advection Diffusion

Varying velocity field vx and vy

$$\partial_t u + (u \cdot \nabla) u = \nu \Delta u$$



Burgers Equation

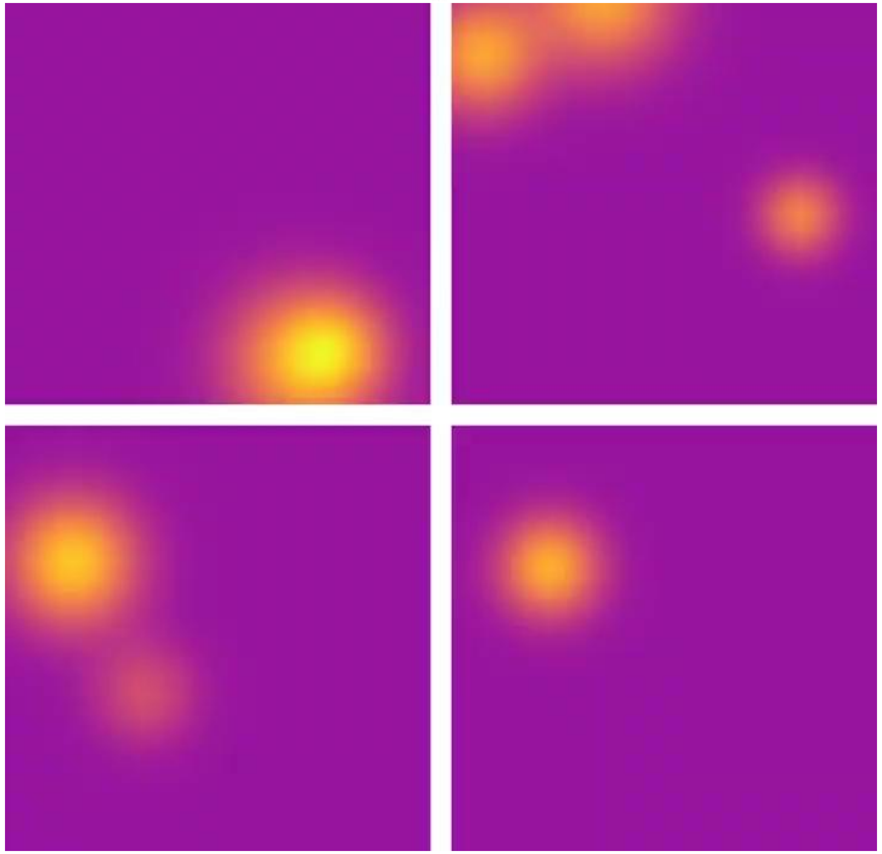
Varying viscosity $\boldsymbol{\nu}$

Navier-Stokes

Varying forcing coefficients

Experimental Benchmarks

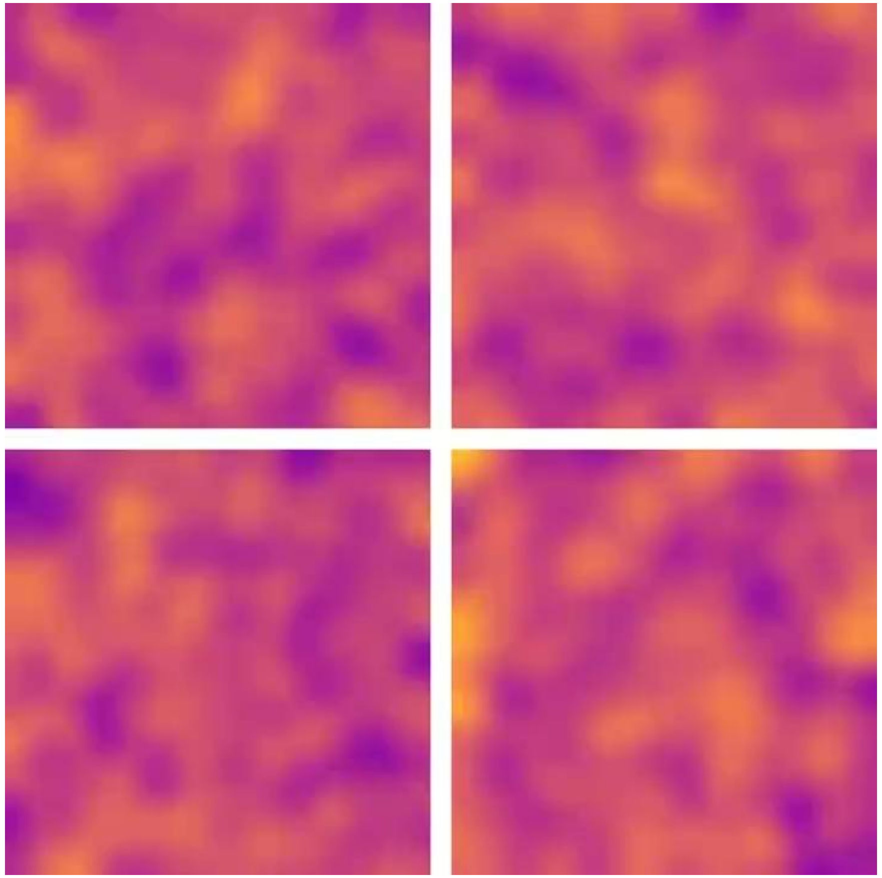
$$\partial_t u = v, \qquad \partial_t v = (c_p^2 - c_s^2)\,\nabla(\nabla\cdot u) + c_s^2 \Delta u,$$



Elastic Waves

Varying P-wave cp and S-wave speeds cs

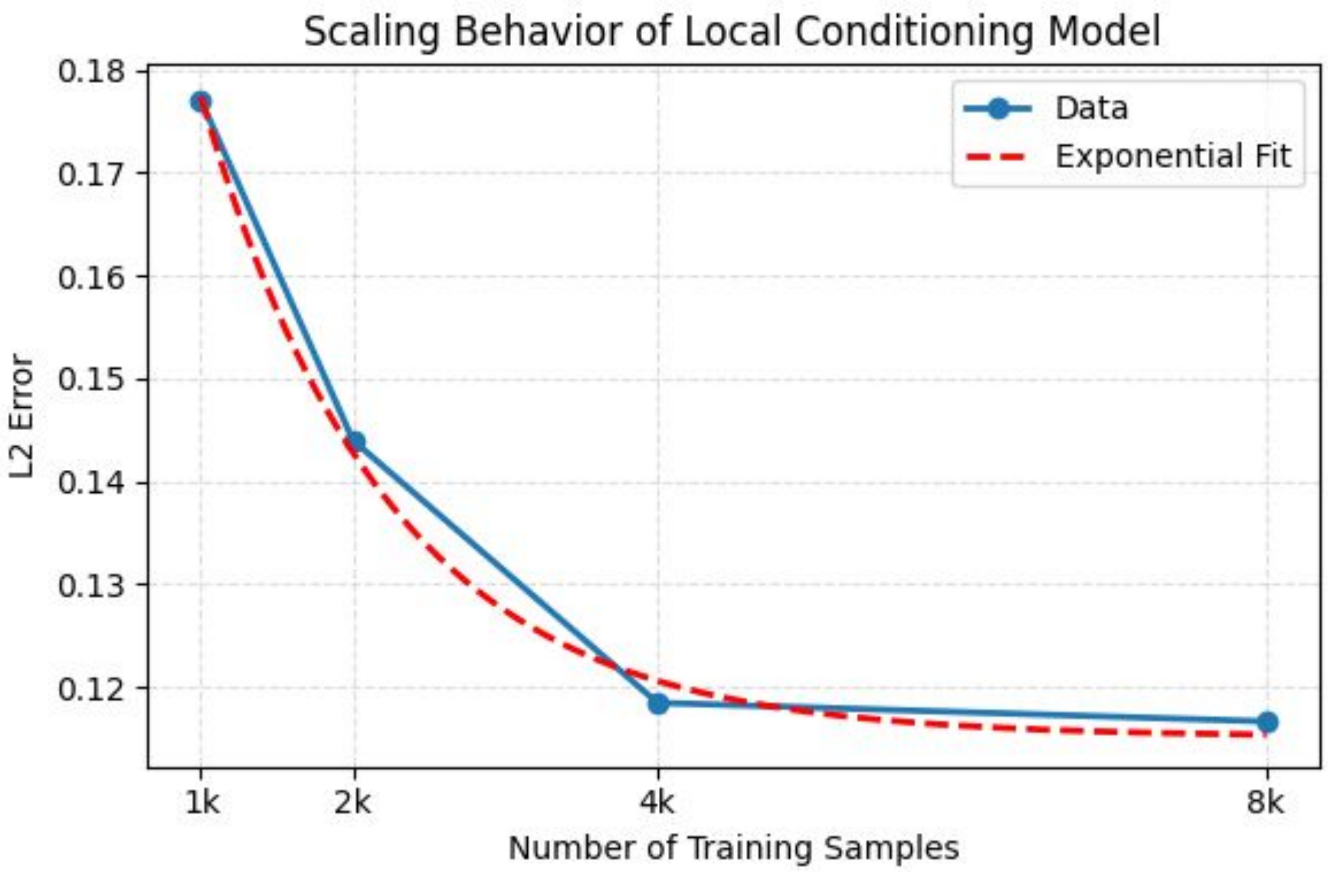
$$\partial_t \omega + u \cdot \nabla\omega = \nu\Delta\omega + f(x,y)$$



Navier–Stokes Equation

Varying external forcing amplitudes

Data Scaling Analysis



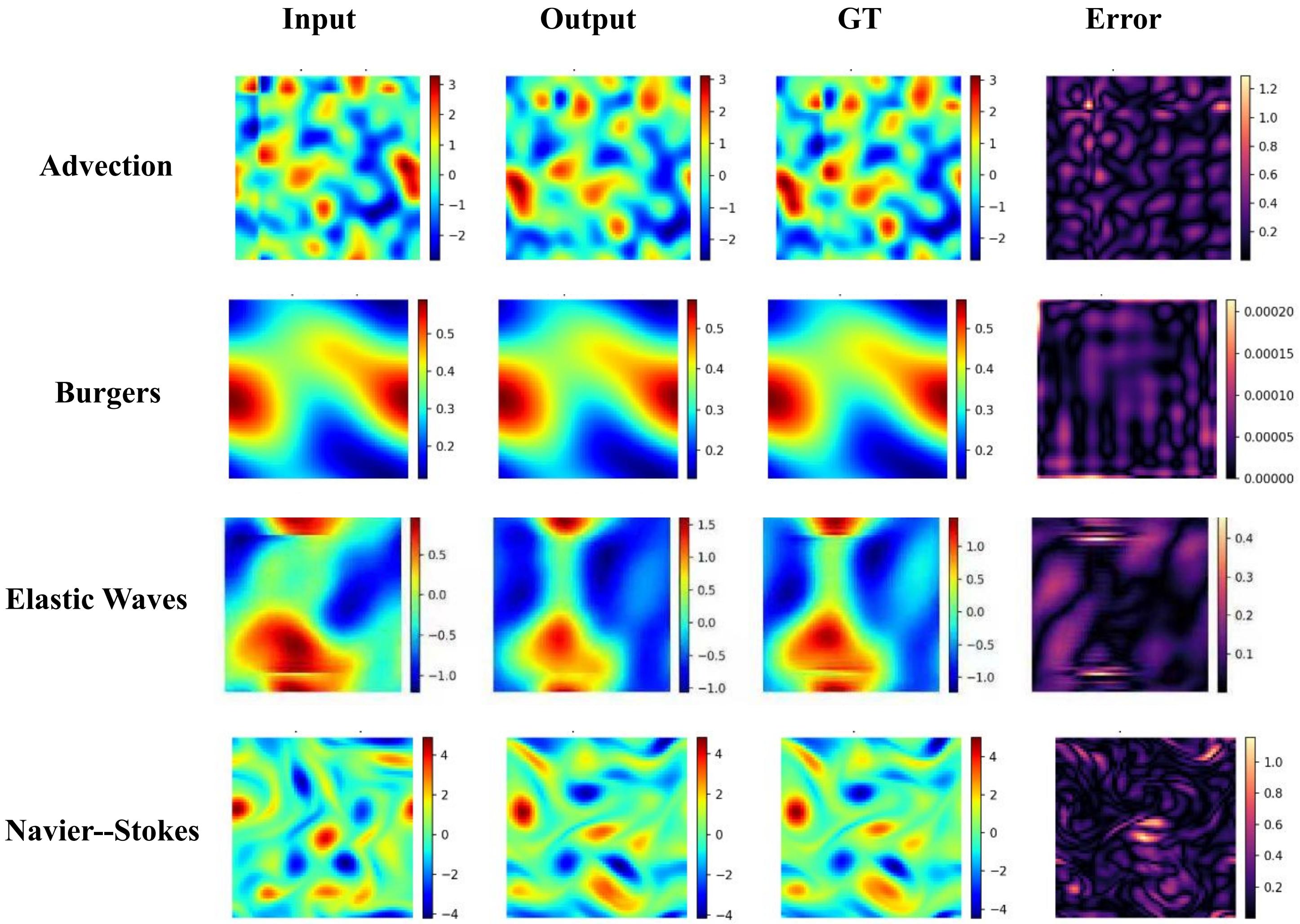
Performance saturation of Local Conditioning on NS equation.
We observe diminishing returns after 4,000 samples, suggesting the model reaches its capacity limit for this specific problem size.

Quantitative Results (Relative L2 Error)

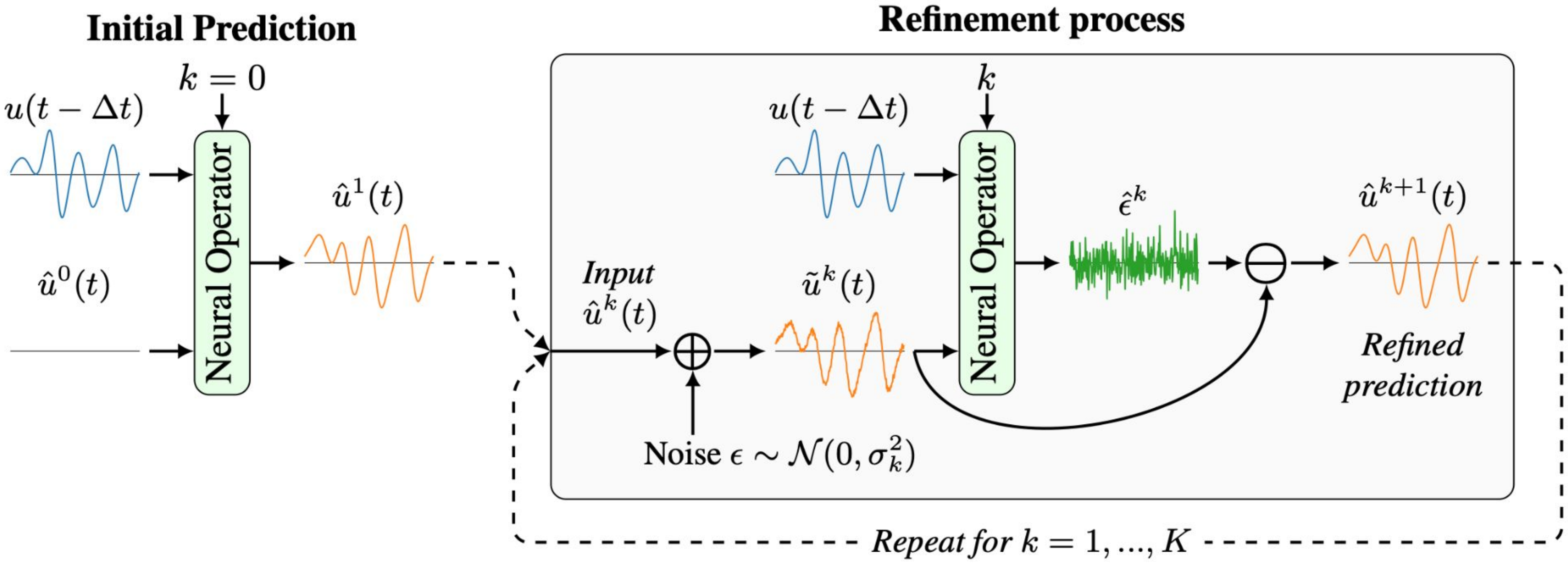
Method	Advection	Burgers	Elasticity	Navier-Stokes
Local Conditioning	**0.1050**	**0.0016**	0.3232	0.1185
Global Conditioning	0.1262	0.0019	**0.2967**	0.1359
Input Concatenation	0.1343	0.0023	0.2979	**0.1175**
Vanilla FNO	0.9430	0.0369	0.5057	0.1490

**Lower is better. Yellow highlights indicate the best performance per task.*

Qualitative Results

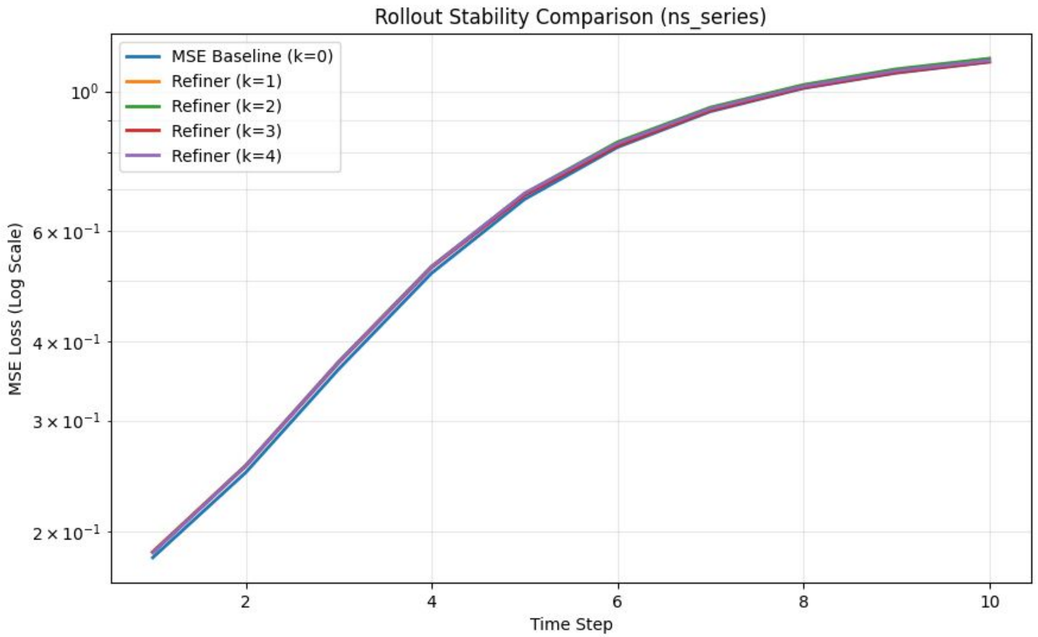


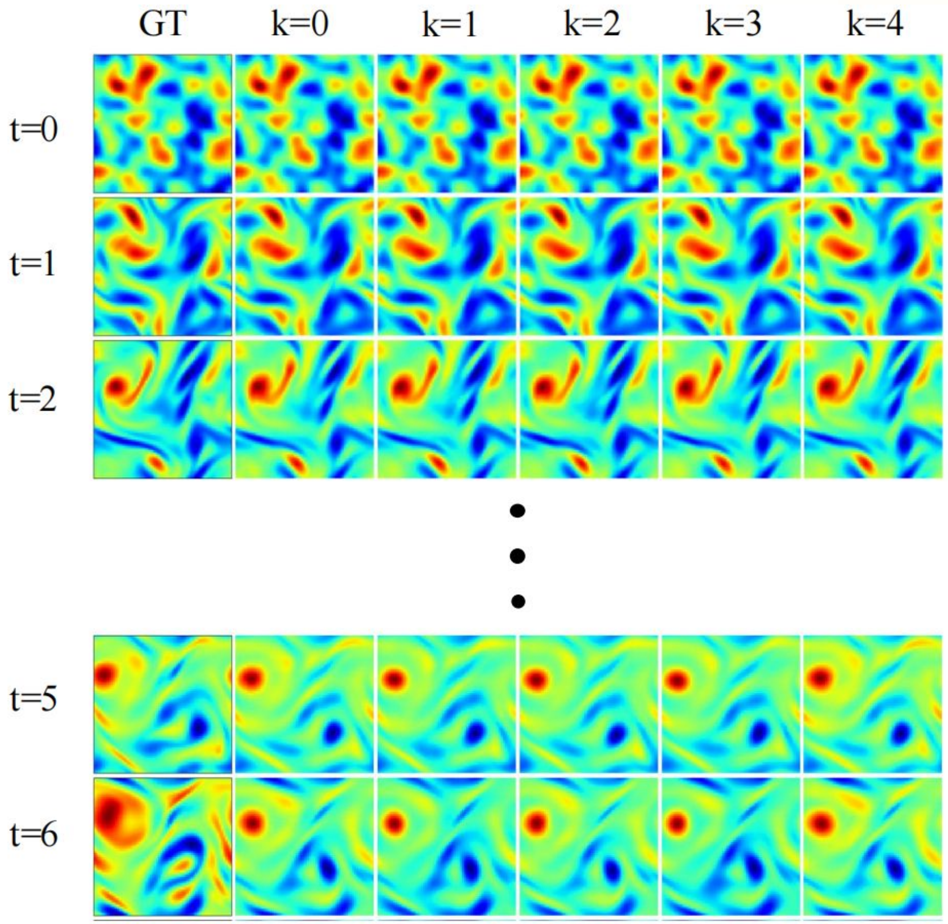
Open Question: Stable Rollout?



Ref:Lippe, P., Veeling, B. S., Perdikaris, P., Turner, R. E., & Brandstetter, J. (2023). *PDE-Refiner: Achieving accurate long rollouts with neural PDE solvers*. arXiv. https://arxiv.org/abs/2308.05732

Refinement model on the Navier–Stokes





Conclusion

- We explore three conditioning strategies for parameterized PDE learning—**local conditioning**, **global conditioning**, and **input-level conditioning**.

- Across four PDE benchmarks, **all conditioned models outperform vanilla FNO**, showing that adding physical parameters is valid for parameter-dependent operator learning.

Thanks for listening!

Analysis: Why different winners?

- **Advection-Diffusion (Local Wins):** This system is highly sensitive to direction (velocity). Local conditioning allows the operator to fundamentally change its "flow" direction in the spectral domain.
- **Elasticity (Global Wins):** Wave speeds change the propagation globally. A simple channel-wise rescaling (Global) captures this frequency-dependent scaling efficiently.
- **Navier-Stokes (Input Wins):** The dominant transport term is parameter-independent. The parameter only affects small-scale dissipation, which the baseline network handles sufficiently well.